Neuro-Computing Techniques for Prediction of Compressive Strength of Concrete

A Thesis
Submitted in Fulfillment of the
Requirements for the Award of the Degree of

Doctor of Philosophy

Submitted by

Palika Chopra
(Registration No. 950911002)

Under the Supervision of

R. K. SHARMA
Professor
Department of Computer Science and Engineering
Thapar University, Patiala

MANEEK KUMAR
Professor
Department of Civil Engineering
Thapar University, Patiala

Thapar University, Patiala

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
Thapar University
Patiala – 147 004 (Punjab) India

January 2017

CERTIFICATE

I, **Palika Chopra**, hereby certify that the work presented in the thesis entitled "**NEURO-COMPUTING TECHNIQUES FOR PREDICTION OF COMPRESSIVE STRENGTH OF CONCRETE**", in fulfilment of the requirement for the award of the degree of **DOCTOR OF PHILOSOPHY** in the Department of Computer Science and Engineering, Thapar University, Patiala, is an authentic record of my own work carried out during the period from July 2010 to January 2017 at this university under the supervision of **Dr. R. K. Sharma** and **Dr. Maneek Kumar**.

The matter presented in this thesis has not been submitted for the award of any other degree in any university.

Palika Chopra.

(PALIKA CHOPRA)

Dated: January 2, 2017

Certified that the above statement made by the candidate is correct to the best of our knowledge.

R.K. Sharma

(R. K. Sharma) 2.1.2017

Professor

Department of Computer Science and Engineering

Thapar University, Patiala-147004

Maneek

(Maneek Kumar)

Professor

Department of Civil Engineering

Thapar University, Patiala-147004

ACKNOWLEDGEMENT

Indeed, my doctorial thesis is a dream of my heavenly father-in-law Sh. Devinder Kumar Chopra, coming true. First and above all, I praise God, the Almighty for proving me this opportunity and granting me the capability to proceed successfully.

Time has provided me the cherished opportunity to express my heartfelt gratitude to my supervisors Dr. R. K. Sharma, Professor and Dr. Maneek Kumar, Professor, Thapar University, Patiala. I shall ever remain indebted to them for their meticulous guidance, constructive criticism, clear thinking, keen interest, constant encouragement and forbearance right from the beginning of this research to its completion. Without their tireless efforts and outstanding knowledge of the subject, it would not have been possible for this research work to take the shape of this thesis. Their association with this quest of mine shall remain a beacon light to me throughout my life.

I would like to acknowledge the most important person - my mother, Smt. Deepa Vohra. It was her determination and constant encouragement that ultimately made it possible for me to see this work through to the end.

I would like to express my deepest gratitude to my family and friends, whose backing of all kinds were not only central to the completion of this thesis, but also the backbone in the ups and downs of my life. Of this group of people, I would like to single out my sister Konika, brother Shukal, my sister-in law Shruti and her husband Vinay, who - as always - deserve my deepest thanks. A special 'thanks' to Heaven Singh for his support during the last stage of this journey.

Of all, my loving husband Tanuj gets the greatest credit, without whose strength, I would not have been able to achieve this pursuit.

Last but not least, I wish to express my thanks to all those who remained behind the screen but whose work has often been consulted and quoted in present research.

Palika Chopra.

(PALIKA CHOPRA)

ABSTRACT

Neuro-computing techniques are being widely used these days and these techniques are considered good for forecasting applications. In recent years, these techniques have also been applied to many civil-engineering problems with reasonable success. These techniques are particularly useful in applications where the complexity of the data or task makes the design of such a function impractical. As the development of concrete strength is a complex non-linear process, depending upon many parameters, it is a problem well suited for applying neuro-computing techniques. In this work, we focused on the prediction of compressive strength of concrete with the help of neuro-computing techniques.

Data for this work has been taken from the experiments conducted by Kumar (2003). For generating a reliable data bank on concrete compressive strength, he considered five parameters, namely, water-cementitious ratio, cementitious content, water content, workability and curing ages in his experiments. He performed all experiments in controlled laboratory conditions. A set of 15 cubes for each of the mixes so proportioned were cast and tested after 28, 56 and 91 days of curing. Thus, an extensive data bank for analyzing the compressive strength of concrete has been used in the present work. Factor analysis has been performed on the data in order to decide variables for predicting compressive strength of concrete with the help of SPSS and investigation, reveal that water-cementitious ratio is the leading predictor variable.

Regression models have been also developed in this work. These have been developed: (i) To analyze the effect of workability on compressive strength; (ii) To analyze the effect of FA on compressive strength; and (iii) For predicting compressive strength of concrete with three different aggregate zones, *i.e.*, Zone-A, Zone-B and Zone-C with and without FA. The regression models developed can predict compressive strength of various mixes very efficiently. However, a variation in data affects the regression coefficients to a large extent. Therefore, we have introduced the ridge parameter in regression equations. By introducing ridge parameter in regression equations, we could obtain more trustworthy and efficient predictive models for compressive strength of concrete that are not affected by the variations in dataset used for prediction.

The development of neural network and GP models for the prediction of compressive strength of concrete has also been undertaken in this work. For the development of an

efficient neural network model, a total of 1440 pilot experiments have been conducted before selecting the final architecture of neural network. As an outcome of this study, it has been inferred that the neural network approach has a great potential for prediction of compressive strength of concrete. The results obtained from neural network model are compared with regression model and GP model. It has been found that neural network models show a high degree of consistency with experimentally evaluated compressive strength of concrete specimens used.

One can further refine the neural network models and GP models proposed in this work by considering a larger and reliable dataset. One can also explore other soft computing models for more efficient prediction of compressive strength of concrete.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

AF	Activation Function
ANFIS	Adaptive Neuro-Fuzzy Inference System
BFGS	Quasi-Newton Backpropagation
BP	Backpropagation
BR	Bayesian Regularizarion
c	Cement content
ca	Coarse Aggregates
ca/*cm*	Coarse aggregate - Cementitious ratio
ca1	Coarse Aggregate of Type - I
ca2	Coarse Aggregate of Type - II
ca3	Coarse Aggregate of Type - III
CC	Cascade Correlation
CGB	Conjugate Gradient Backpropagation with Powell-Beale Restarts
CGF	Conjugate Gradient Backpropagation with Fletcher- Reeves Update
CGP	Conjugate Gradient Backpropagation with Polak-Ribiere Update
cm	Cementitious content
ESIM	Evolutionary Support Inference Model
fa	Fine aggregates
fa/*cm*	Fine aggregate - Cementitious ratio
FA	Fly Ash
f_c	Compressive strength of concrete
f_{c28}	Compressive strength of concrete at curing age of 28 days
f_{c56}	Compressive strength of concrete at curing age of 56 days
f_{c91}	Compressive strength of concrete at curing age of 91 days
$f_{c91,28}$	Compressive strength of concrete at curing age of 91 days and f_{c28} is engaged as one of the independent parameter
$f_{c91,56}$	Compressive strength of concrete at curing age of 91 days and f_{c56} is engaged as one of the independent parameter
f'_{c28}	Compressive strength of concrete at curing age of 28 days, when cement is replaced by 15% of FA
f'_{c56}	Compressive strength of concrete at curing age of 56 days, when cement is replaced by 15% of FA
f'_{c91}	Compressive strength of concrete at curing age of 91 days, when cement is replaced by 15% of FA
FL	Fuzzy Logic
FPNN	Fuzzy Polynomial Neural Network
GD	Gradient Descent
GDM	Gradient Descent with Momentum
GMDH	Generalized Group Method of Data Handling
GP	Genetic Programming
goal	Network performance goal

GWPOT	Genetic Weighted Pyramid Operation Tree
hn	Hidden Neuron
HSC	High Strength Performance
HPC	High Performance Concrete
lamda	Number of children produced
LM	Levenberg-Marquardt Backpropagation
lr	Learning Rate
LWC	Light Weight Concrete
MAE	Mean Absolute Error
MART	Multiple Additive Regression Tree
MED	Medium
MPa	Mega Pascal
MSE	Mean Square Error
mu	Population Size
OSS	One Step Secant Backpropagation
R	Correlation
R^2	Coefficient of determination
RMSE	Root Mean Square Error
RP	Resilient Backpropagation
RPE	Relative Percentage Error
SCC	Self Compacting Concrete
SCG	Scaled Conjugate Gradient Backpropagation
SD	Standard Deviation
SF	Silica Fume
SFRC	Steel Fibre Reinforced Concrete
SPSS	Statistical Package for the Social Sciences
SSD	Standard Surface Dry
SVM	Support Vector Machine
w	Water
w/*cm*	Water-Cementitious ratio

CHAPTER 1

NEURO-COMPUTING PARADIGM AND ITS ENGINEERING APPLICATIONS: AN INTRODUCTION

1.1 Introduction

The field of 'neural networks' (also known by varied names as neuro-computing, artificial neural networks and computational neural networks) is a branch of cognitive science that has originated from diverse sources, ranging from the enthrallment of mankind with understanding and emulating the human brain, to broader issues of copying human abilities such as speech and the use of language, to the practical, commercial, scientific, and engineering disciplines of pattern recognition, modeling, and prediction. The pursuit of technology is a strong driving force for researchers, both in academia and industry, in many fields of science and engineering. In neural networks, as in 'machine learning', the excitement of technological progress is supplemented by the challenge of reproducing intelligence itself. A broad class of techniques can be categorized in this heading. In general, networks consist of layers of interconnected neurons, each neuron producing a nonlinear function of its input. The input to a neuron may come from other neurons or directly from the input data. Also, some neurons are identified with the output of the network. The complete network, therefore, represents a very complex set of interdependencies which may incorporate any degree of nonlinearity, allowing very general functions to be modeled. In the simplest networks, the output from one neuron is fed to another neuron in such a way as to propagate the inherent features through layers of interconnecting neurons. It has been argued that neural networks exemplify to a certain extent the behavior of networks of neurons in the brain. Neural network models combine the complexity of statistical techniques with machine learning to reproduce human intelligence; however, all have been done at 'unconscious' level and therefore, it is opaque to users and there is no method available to provide evidence about learned perceptions to its users.

The application of neuro-computing techniques in the field of civil engineering started in early nineties and has since encompassed almost all allied fields of civil engineering, namely, environmental engineering, construction engineering and management, geotechnical engineering

and structural engineering. Concrete, with its innovations and variations, is widely used for constructing most of structures and forms the backbone of civil engineering. Its service is considered to be tremendously long, exclusive of the disaster caused by earthquakes. Concrete is an ubiquitous building material that has been in use for millennia. It is being widely used for building roadways, bridges, buildings, walkways, and numerous other structures. Conventional concrete is a mixture of water, Portland cement, fine and coarse aggregates. Supplementary materials such as chemical and mineral admixtures may be added to the basic concrete ingredients to enhance its properties in fresh or hardened state. The process of selecting suitable ingredients for concrete and their relative amount with objective of producing concrete of required strength, durability and workability as economically as possible is termed as mix design.

The earlier concrete manufacturers used to employ a variety of concrete mix designs having variable strengths, slumps and other properties, which were optimized through trial and error testing and/or were based on standard mix design tables. But, given the extremely large number of possible concrete mix designs, coupled with the practical inability to test even a small fraction of such mix designs, the likelihood of identifying the most "optimized" mix design through these procedures is quite low. Thus, development of tools to find the optimized mix proportions has been the subject of research for more than the last four decades. The objective of any proportioning method is to determine an adequate and economic rate for the materials making-up the concrete, which can be used in its production, giving mix properties as close as possible the desired properties, with the lowest cost.

The engineering properties of cement-based materials and special concretes depend on various parameters including the non-homogeneous nature of their components, the intrinsically different properties of various elements and sometimes on the twin and/or contradictory effects of some ingredients on the overall concrete performance. Therefore, a clear understanding of such complex behavior is needed in order to successfully use these materials in various engineered structures. An effort, in the present research, has been made to develop neural networks for prediction of concrete properties. The following sections elaborate the various elements of the biological neural network and also the artificial neural network.

1.2 Biological neural networks

A neuron's dendritic tree (Fig. 1.1) is connected to thousands of adjoining neurons. When one of those neurons fires, a positive or negative charge is received by one of the dendrites. The strengths

of all the received charges are added together through the processes of spatial and temporal summation. Spatial summation occurs when several weak signals are converted into a single large one, while temporal summation converts a rapid series of weak pulses from one source into one large signal.

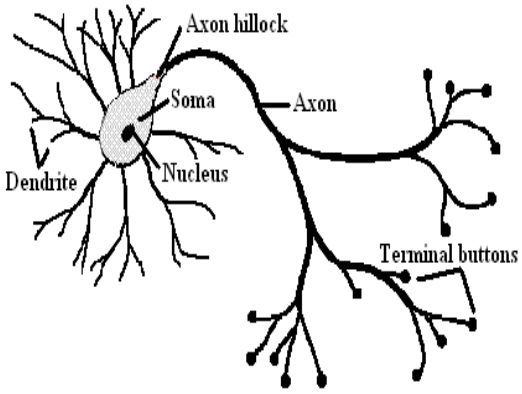

Fig. 1.1: Schematic of biological neuron

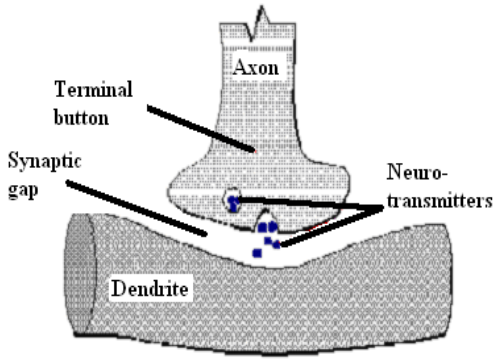

Fig. 1.2: The synapse

The Soma (Cell body) will receive the aggregated input and if it is greater than threshold value of Axon Hillock, then neuron will fire and an output signal is transmitted to the axon. Each terminal button will receive the output of the same intensity and it is unaffected by the many divisions in the axon. Synapse will help to connect each terminal button to other neurons (Fig. 1.2).

1.3 Artificial neural network

Artificial neural networks (called neural networks now onwards) consist of the following three primary elements (Simpson, 1996):

i. **Topology** – the organization of layers and interconnection between these layers;

ii. Learning – the methods to store information in the network; and

iii. Recall – the retrieval of stored information from the network.

The basic arrangement of a neural network consists of artificial neurons (Fig. 1.3). The neurons are also known as processing elements (PEs), nodes, neurodes, units, *etc.*, and are similar to biological neurons in the human brain, which are arranged into layers.

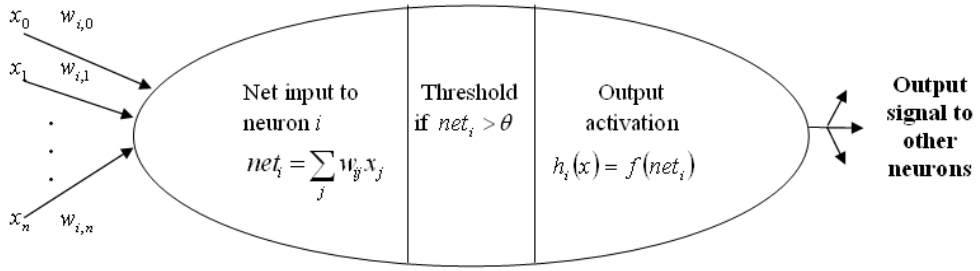

Fig. 1.3: Schematic representation of general neuron model

The most common neural network structure consists of an input layer, one or more hidden layers and an output layer (Knight, 1990). The neural networks considered for empirical investigation in this study are also one hidden layer networks with a single output. For elaboration purposes, let the input dimension be $n\ (n \in Z_+)$ and let the number of hidden neurons be $m\ (m \in Z_+)$. The training pairs are represented by $D = \{\mathbf{x}^{(p)}, t^{(p)}\}$, where $p = 1, 2, ..., P;\ P \in Z_+$, is the number of training exemplars; and the index p is always assumed to be present implicitly. The matrix $\boldsymbol{w}$ denotes the input to the hidden neurons connection strength, w_{ij} is the $(i, j)^{\text{th}}$ element of the matrix $\boldsymbol{w}$ representing the connection strength between the j^{th} input and the i^{th} hidden layer neuron. With this nomenclature, the net input to the i^{th} hidden layer neuron is given by

$$net_i = \sum_{j=1}^{n} w_{ij} x_j + \theta_i^{(1)} = \mathbf{w}_i \cdot \mathbf{x} + \theta_i^{(1)} \qquad (1.1)$$

where $\theta_i^{(1)}$ is the bias of the i^{th} hidden layer neuron. The output from the i^{th} hidden layer neuron is given by

$$h_i(\mathbf{x}) = f^{(1)}(net_i) \qquad (1.2)$$

where $f^{(1)}(\cdot)$ is a nonlinear activation function.

The equation for the network output neuron is given by

$$net_o = f^{(2)}(net) = net \tag{1.3}$$

as $f^{(2)}(\cdot)$ has been considered as a linear function in this study.

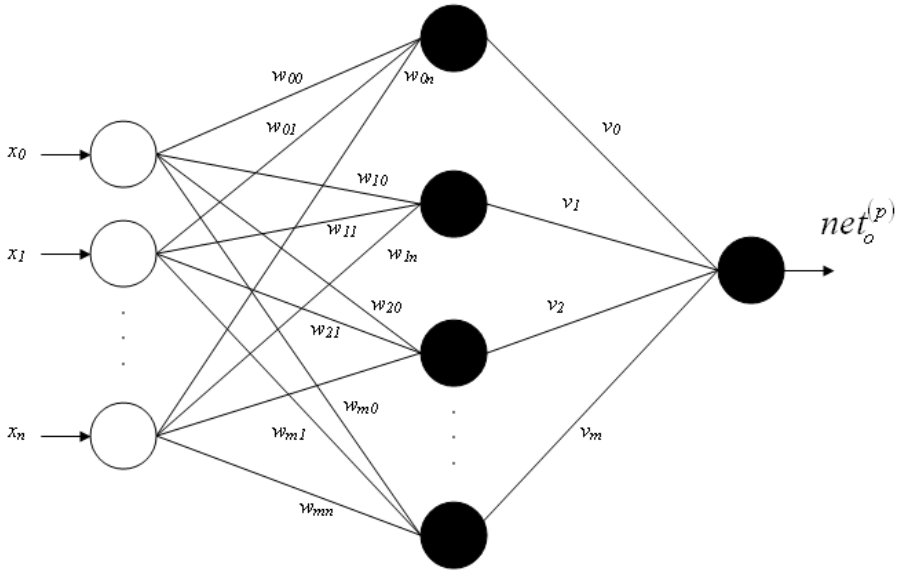

Fig. 1.4: Schematic of feed forward neural network models

The notations are presented diagrammatically in Fig. 1.4. This figure represents an *n*-input neurons, *m*-hidden and *1*-output neuron feed forward neural network. Such a neural network is trained to fit a dataset *D* by minimizing an error function (or performance function) as

$$F = E_D(\mathbf{W}) = \frac{1}{P}\sum_{p=1}^{P}\varepsilon^2 = \frac{1}{P}\sum_{p=1}^{P}\left(net_o^{(p)} - t^{(p)}\right)^2 \tag{1.4}$$

This function is minimized using standard optimization method, the elements of **w** and **v** are updated accordingly.

1.4 Neural network learning

Learning in the present perspective may be defined as a modification in connection weight values that effect in the information capture that can later be retrieved. Generally, the initial weights for the network earlier to training are set to random values within a pre-defined range. This method is used widely in error-correction learning systems that are mostly used in civil engineering

applications. The following learning methodologies are generally adopted for neural network training.

i. **Supervised learning** – For supervised learning, the neural network need to be trained using many samples. Each sample contains input values with its corresponding output value. The network will compute the desired output by continuously updating the weights and minimizing the error in each epoch. This process is known as training of the network. The supervised learning approach is followed in the present study.

ii. **Unsupervised learning –** It is self-supervised learning and it does not require output values for network training. Each sample input corresponds to some distinct class and this training process needs to uncover these classes.

1.5 Network architecture

Neural network architecture (or network topology) refers to the types of interconnections between neurons. A network is said to be fully connected if the output from a neuron is connected to every other neuron in the next layer. A network with connections that passes outputs in a single direction only to neurons on the next layer is called a feedforward network. A feedback network allows its outputs to be inputs to preceding layers. Networks that work with closed loops are known as recurrent networks. Feedforward networks are faster than feedback networks as they require a single pass only to obtain a solution. The former networks are most commonly used for prediction problems (Mukherjee and Deshpande, 1995). Feedback networks being time-variant models are not pertinent to this study. Hence, further discussion is confined only to the feedforward networks.

1.5.1 Neural network parameters

There are numerous network internal parameters allied to network architecture and training methods that need to be adjusted on experimental basis to optimize the network performance (Rumelhart *et al.,* 1994; Refenes, 1995; Ghedira and Bernier, 2004). A brief description of the major parameters is given below.

i. **Number of inputs and outputs:** The number of inputs to the network and the number of outputs from the network are always problem specific. It depends on the number of variables associated with the problem statement.

ii. **Data partitioning:** The purpose of the data partitioning is to randomly divide the input data set into separate data sets *i.e.,* training data set, validation data set or testing data

set. In neural networks, there are no standard methods that can perform precise allocation of the data set. So, data partitioning is performed by keeping an adequate balance between the data sets where there is enough data for training, while at the same time leaving sufficient data behind for validation or testing. Choosing optimal fraction of data points for training and for validation sets is a research issue (Coakley, 1991; Guyon and Elisseeff, 2003).

iii. **Number of hidden layers:** The number of hidden layers is ascertained by 'trial and error' method in each application and is of prime importance. The increment in the number of hidden layers increases the accuracy to a great extent, but, alongside, the complexity and training time of the network is also increased many fold (Karsoliya, 2012).

iv. **Number of hidden neurons:** The stability of the network is measured on error basis and the number of hidden neurons influences the error on the nodes to which their output is connected directly. The problem of over-fitting occurs, if the excessive numbers of hidden neurons are used. So, it has always been a research issue to find the optimal number of hidden neurons. Many researchers have already worked on it (Li *et al.,* 1995; Tamura and Tateishi, 1997; Fujita, 1998; Zhangi, 2003; Jinchuan and Xinzhe, 2008; Xu and Chen, 2008; Shibata and Ikeda, 2009; Hunter *et al.,* 2012). Recently, Sheela and Deepa (2013) have argued that we may consider the number of hidden neurons in a neural network as $(4n^2 + 3)/(n^2 - 8)$, where n is the number of input neurons. The proposal was to devise the number of hidden neurons as a function of input neurons *(n)*.

v. **Learning rate (*lr*):** Learning rate is used to control the amount of weight adjustment at each step of training. The learning rate, ranging from 0 to 1, determines the rate of learning at each time step. Small values of the learning rate increase learning time but tend to decrease the chance of overshooting the appropriate optimal solution. Large values of the learning rate may train the network faster, but may result in no learning taking place at all (Hassoun, 1995; Keckman, 2001).

vi. **Training and testing tolerances:** The training tolerance is the amount of accuracy that the network is required to achieve during its learning stage on the training dataset. The

testing tolerance is the accuracy that will determine the predictive strength of the neural network on the test dataset.

vii. **Initial weights:** The ultimate solution may be affected by the initial weights of a multilayer feed-forward network. They are initialized at small random values. The choice of the initial weight determines how fast the network converges. A number of alternative suggestions have been based on assigning decision boundaries with positions and orientations defined by the training examples (Lee *et al.,* 1991; Wessels and Barnard, 1992; Bishop, 1995; Hassoun, 1995; Sumpter and Donald, 1996; Atiya and ji, 1997; Kecman, 2001; Masood, 2012; Sutskever *et al.,* 2013).

viii. **Activation functions (AF):** The activation function helps to calculate an output from the summation of weighted inputs and often non-linear activation function is used at the hidden layer. The choice of activation function has always a great influence on the performance and complexity of network (Duch and Jankowski, 1999). Sigmoidal activation function is the most commonly used activation function. A number of AFs have been investigated by researchers (Moody and Yarvin, 1992; Duch and Jankowski, 1999). The employ of new activation functions is an open research activity that can fabricate exciting results in different applications (Skoundrianos and Tzafestas, 2004). The most common activation functions are represented in Table 1.1.

Table1.1: Common activation functions

S. No.	Activation Functions	
	Name	**Description**
1	tan-sigmoid	$tansig(x)=\frac{2}{1+e^{-2x}}-1$
2	log-sigmoid	$logsig(x)=\frac{1}{1+e^{-x}}$
3	Purelin	$purelin(x)=(x)$

ix. **Back-propagation training algorithms:** The term back-propagation refers to the process by which derivatives of the network error, with respect to network weights and biases, can be computed. This process can be used with a number of different optimization strategies. There are several different back-propagation training algorithms. Each of these algorithms has a variety of different computation and storage requirements, represented in Table 1.2. The training algorithms used in this study are briefly described below.

- **Gradient Descent Backpropagation (GD):** The GD backpropagation algorithm is based on minimizing MSE between the output of network and target output. The network will be considered converged and trained when the error will decreased to the specified threshold level.
- **Gradient Descent with Momentum Backpropagation (GDM):** In GDM algorithm, momentum acts like a low-pass filter. It allows network to respond to recent trends and ignore small features in error surface. This is not possible without momentum as network may get stuck itself in shallow local minimum.
- **Gradient Descent with Adaptive Learning Rate Backpropagation (GDA):** In GD algorithm, network is used the constant learning rate (*lr*) while GDA helps it to be adaptive and as large as possible while keeping it stable.
- **Resilient Backpropagation (RP):** RP algorithm updates weights according to error function behaviour with local adaptation. It is generally faster but requires some increase in memory requirement.
- **Conjugate Gradient Backpropagation with Fletcher-Reeves Update (CGF):** In CGF algorithm, conjugate direction is the direction of search which helps this algorithm to be faster. The new search direction is determined as to conjugate to last search directions.
- **Conjugate Gradient Backpropagation with Polak-Ribiére Update (CGP** In CGP algorithm, next search direction is the inner product of last change in gradient with current gradient divided by norm squared of last gradient. The CGP uses four vectors for storage and CGF uses three vectors. Therefore, storage requirements of CGP are slightly higher than CGF.
- **Conjugate Gradient Backpropagation with Powell-Beale Restarts (CGB):** In all discussed above conjugate gradient algorithms, when the number of iterations is equal to the number of network parameters, then search direction is reset to negative of the gradient. But in CGB algorithm, it restarts if there is very small orthogonality seen between the current and last gradient. It uses six vectors, so it works on largest storage requirements than other conjugate gradient algorithm.
- **Scaled Conjugate Gradient Backpropagation (SCG):** All three conjugate gradient algorithm, discussed so far, are very time consuming since network

need to response to every input several times for individual search. This also increases the computational expensive. The SCG algorithm reduces computations in each epoch. It uses only three vectors as storage requirement.

- **BFGS Quasi-Newton Backpropagation (BFGS):** BFGS can be best training function for small networks as it is faster than conjugate gradient methods. Since the basic step is to form Hessian Matrix for network which is complex and expensive.
- **One Step Secant Backpropagation (OSS):** The OSS uses less storage and less computation per iteration than BFGS and uses little more storage than conjugate gradient algorithms.
- **Levenberg-Marquardt Backpropagation (LM):** LM algorithm is basically good when high precision is required. It has larger computational requirements for each epoch. It converges better than any other BP method and uses $O(N^2)$ storage where N is the total number of weights.
- **Bayesian Regularization Backpropagation (BR):** BR algorithm is basically the best approach to overcome the problem of over-fitting of neural networks and helps to enhance the prediction accuracies for unseen data. It uses the same function for updates of weights as used by the LM algorithm.

Table 1.2: List of training algorithms and their brief description

Algorithm	Description
traingd()	Basic gradient descent. It is slow in response and can be used in incremental mode training.
traingdm()	Gradient descent with momentum. It is generally faster than traingd and can be used in incremental mode training.
traingda()	Adaptive learning rate. It will train the network faster than traingd, but it can only be used in batch mode training.
trainrp()	Resilient back-propagation (RP) algorithm. It is the simple batch mode training algorithm with convergence and minimal storage requirements.
traincgf()	Fletcher-Reeves conjugate (CGF) algorithm. It has smaller storage requirements than CGP and CGB.
traincgp()	Polak-Ribiere conjugate gradient (CGP) algorithm. It has faster convergence on some problems, but has larger storage requirements.
traincgb()	Powell-Beale conjugate gradient (CGB) algorithm. Generally, it converges very fast and has slightly larger storage requirements.
trainscg()	Scaled conjugate gradient (SCG) algorithm. It is the only conjugate gradient algorithm that requires no line search. A very good general purpose training algorithm.

trainbfg()	BFGS quasi-Newton method (BFG) algorithm. It requires storage of appropriate Hessian matrix and has more computation, in each epoch, than conjugate gradient algorithms, but usually converges in less iteration.
trainoss()	One step secant (OSS) algorithm. It requires less storage than BFGS and little more than conjugate algorithms.
trainlm()	Levenberg-Marquardt (LM) algorithm. It is the fastest training algorithm for the network of moderate size. It has memory reduction feature for the use when the training set is large.
trainbr()	Bayesian regularization. It is the modification of the Levenberg-Marquardt training algorithm to produce networks that generalize well. It reduces the difficulty of determining the optimum network architecture.

1.6 Crafting a neural network

The process to devise a neural network model involves the following steps.

- **i.** For supervised neural network, the dataset should be with both input values and an output value.
- **ii.** The dataset will be divided into training and testing datasets. Training dataset is used in the learning process of the model and test dataset is used to validate the model's predictive abilities.
- **iii.** The neural network architecture will be defined before starting its learning process.
- **iv.** A backpropagation algorithm will be used to adjust the weights to minimize the error.
- **v.** An evaluation process is carried out to determine the network's predictive abilities and if an acceptable level of accuracy is obtained then it is considered ready to use.
- **vi.** A large number of models with different parameters and architectures should be developed before determining an acceptable model since there are no fixed rules available for the development of neural network. If a neural network is over trained, then it will not be able to give generalized output and starts fit itself to the training set only. This could result in poor predictions of test datasets.

These steps are depicted as a flowchart in Fig. 1.5.

1.7 Measure of performance evaluation for neural network models

For the success of a supervised learning algorithm, it is always necessary to have some reliable quantitative learning measurement method. Root mean squared error (RMSE) is an adequate and commonly used for error measurements. RMSE is used to find out how close a network is in its predictions with respect to the target output values. RMSE values should be less than 0.1. This

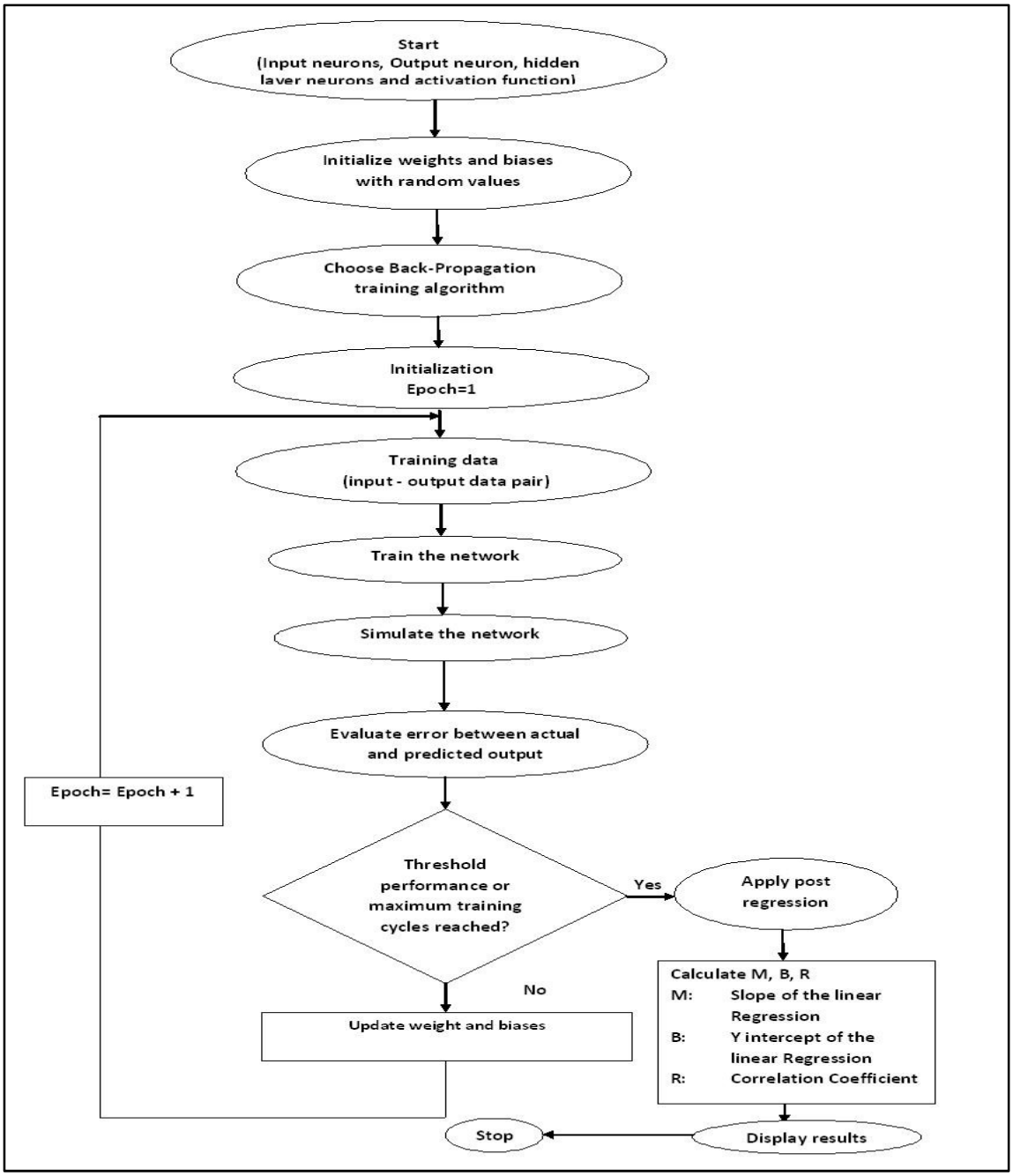

Fig. 1.5: A plot to demonstrate the algorithm of neural network model

value indicates a prediction accuracy of 90% and it means that a network has learned from its training set (Dayhoff, 1990; Wasserman, 1993; Grzesiak *et al.*, 2006). RMSE is calculated using Eq. (1.5) and Eq. (1.6).

$$\text{Prediction error} = \frac{\text{Predicted value} - \text{Actual value}}{\text{Actual value}} \times 100\,\% \qquad (1.5)$$

$$\text{RMSE} = \sqrt{\frac{\sum_{p=1}^{P} (\text{Prediction error})^2}{P}} \tag{1.6}$$

where P is the number of input patterns.

1.8 Genetic Programming

Genetic programming (GP) has started of as an endeavor to discover as to how computers could be trained to solve problems without being explicitly programmed to do so. It is an extension to genetic algorithms proposed by Koza (1992) who defines GP as "a domain independent problem solving impend in which computer programs are evolved to solve, or approximately solve, problems based on Darwinian principle of reproduction and analogs of naturally occurring genetic operations such as reproduction, crossover and mutation". The five major prelude steps for the basic version of GP require the user to specify: a) the set of terminals for each stem of the to-be-evolved program; b) the set of primitive functions for each stem of the to-be-evolved program; c) the fitness measure; d) certain parameters for controlling the run; and e) the termination criterion and method for designating the results of the run.

1.8.1 Steps to execute Genetic Programming

Genetic programming is initiated with a population of randomly generated computer programs composed of the available programmatic ingredients. It iteratively transforms a population of computer programs into a new generation of the population by applying analogs of naturally occurring genetic operations. These operations are applied to individual(s) selected from the population. The individuals have been elected to participate in the genetic operations based on their fitness. The iterative transformation of the population is implemented inside the main generational loop of the run of genetic programming. The execution steps of GP, as also provided in the flowchart (Fig. 1.6) are as follows:

i) Generation 0 (initial population of individual computer programs) will be created randomly and it is composed from available functions and terminals.

ii) The following sub-steps need to perform on the population iteratively while a termination criterion is satisfied:

a) Execute each program and determine its fitness using defined fitness measurement methods.

b) To participate in the genetic operations in sub-step c, select one or two individual program(s) based on fitness.

c) Create new individual program(s) by applying the following genetic operations:

- Reproduction: The selected individual program will be copied to the new population.
- Crossover: Chose parts of two selected programs randomly and create new program(s) (offspring) for new population by process of recombining.
- Mutation: Chose part of one selected program randomly and mutate it to create new offspring program and add it to new population.
- Architecture-altering operations: Apply architecture-altering operation to one selected program by choosing such operations from the available range of operation and create a new offspring program for new population.

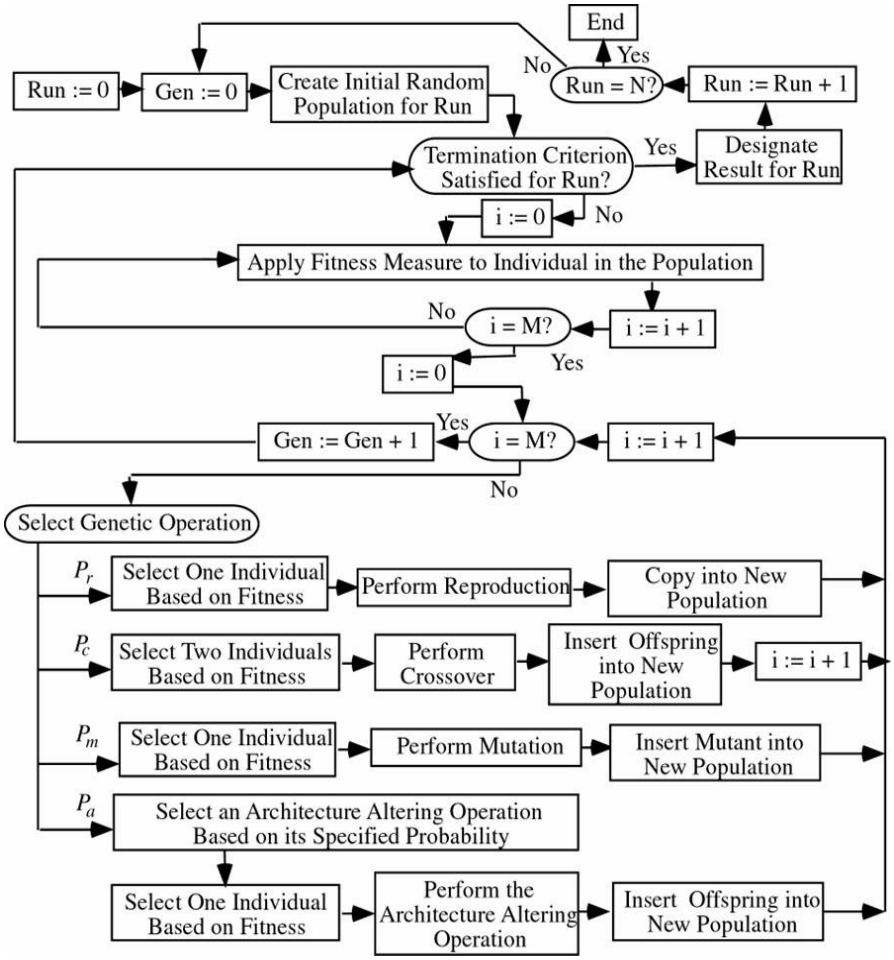

Fig. 1.6: Genetic Programming Flow Chart (Koza, 1992)

iii) If the termination criterion is satisfied, select the best program from the population based on fitness measurement function.

1.9 Concrete mix design

Concrete mix design is termed as the process of selection of suitable ingredients of concrete and estimating their quantities (generally expressed in kilograms per cubic meter) with the objective of producing a concrete of required strength, durability, and workability as economically as possible. The proportioning of the various ingredients of concrete is governed primarily by the performance expected from concrete in two states, namely the plastic state and the hardened state. If the freshly prepared plastic concrete is not sufficiently workable, it cannot be properly placed and compacted. The property of workability is therefore, of vital importance.

The compressive strength of hardened concrete is considered to be one of the most important properties of concrete. It depends on many factors, namely, quality and quantity of cement, water and aggregates; batching and mixing methods; and the techniques used in placing, compaction and curing. The cost of concrete is made up of the cost of materials, plant and labour. The aim of concrete mix design is thus to produce as lean a mix as possible satisfying the strength and durability requirements as discussed above), since the cement is several times costlier than the aggregate. Further, from a technical point of view, the rich mixes (with higher cement content) result in the evolution of high heat of hydration in mass concrete which may cause significant cracking and high shrinkage losses in concrete, thus severely reducing its durability.

The actual cost of concrete is related to the cost of materials required for producing a minimum mean strength called characteristic strength that is specified by the designer of the structure. This depends on the quality control measures, which further adds to the cost of concrete. The extent of quality control methods employed, and hence the corresponding increase in the cost, depends on the size and type of job. The workability of mix may also influence the cost of labour and equipments, *e.g.,* a concrete mix of inadequate workability may require extra labour and/or specialized equipment to achieve a certain degree of workability necessary for the job.

1.9.1 Requirements of concrete mix design

The requirements which form the basis of selection and proportioning of mix ingredients are as follows:

- A minimum compressive strength specified in accordance with the structural consideration;
- An adequate workability of the mix necessary for full compaction with the compacting equipment available;
- maximum water-cement ratio and/or minimum cement content to provide adequate durability for particular site conditions; and
- maximum cement content to avoid shrinkage cracking due to temperature cycles in mass concrete.

1.9.2 Types of concrete mixes

- **Nominal mixes**

  In past the specifications for achieving a particular grade of concrete prescribed the fixed proportions of cement, fine and coarse aggregates. These mixes of fixed cement-aggregate ratio which ensures adequate strength are termed as nominal mixes. These proportions offer a standard mix design, which under the normal circumstances, have a margin of strength above that specified concrete grade. Even today, these mixes may be used for small jobs or temporary construction works. However, due to the variation in the quality of ingredients, source of aggregates, mixing and placing techniques etc., the nominal concrete for a given workability varies widely in strength. Hence, these mixes are generally not preferred for major construction works.

- **Standard mixes**

  The nominal mixes of fixed cement-aggregate ratio (by volume) vary widely in strength and may result in under- or over-rich mixes. For this reason, the minimum compressive strength has been included in many specifications. These mixes are termed standard mixes. IS 456-2000 has designated the concrete mixes into a number of grades as M10, M15, M20, M25, M30, M35 and M40. In this designation the letter M refers to the mix and the number to the specified 28 day cube strength of mix in N/mm^2. The mixes of grades M10, M15, M20 and M25 correspond approximately to the mix proportions (cement: fine aggregate: coarse aggregate) (1:3:6), (1:2:4), (1:1.5:3) and (1:1:2) respectively.

- **Design mixes**

  In these concrete mixes the performance of the concrete is specified by the designer but the mix proportions are determined by the producer of concrete, with a constraint that

minimum cement content has to be provided. This is most rational approach to the selection of mix proportions with specific materials in mind possessing more or less unique characteristics. The approach results in economical production of concrete with most appropriate properties consistent with the ingredients provided for a design. For the concrete with undemanding performance, nominal or standard mixes (prescribed in the codes by quantities of dry ingredients per cubic meter and by slump) may be used only for very small jobs, when the 28-day strength of concrete does not exceed 30 N/mm^2.

1.9.3 Factors affecting the choice of mix proportions

- **Compressive strength**

  It is one of the most important properties of concrete that also influences many other describable properties of the hardened concrete. The mean compressive strength required at a specific age, usually 28 days, determines the nominal water-cement ratio of the mix. According to Abrams' law the strength of fully compacted concrete is inversely proportional to the water-cement ratio.

- **Workability**

  The degree of workability required depends on three main factors. This includes the size of the section to be concreted, the amount of reinforcement, and the method of compaction to be used. For the narrow and complicated section with numerous corners or inaccessible parts, the concrete must have a high workability so that full compaction can be achieved with a reasonable amount of effort. This also applies to the embedded steel sections.

- **Durability**

  The durability of concrete is its resistance to the aggressive environmental conditions. High strength concrete is generally more durable as compared to low strength concrete. In the situations when the high compressive strength is not necessary but the conditions of exposure demand high durability, the durability requirement will govern the water-cement ratio to be used.

- **Maximum nominal size of aggregate**

  In general, larger the maximum size of aggregate, smaller is the cement requirement for a particular water-cement ratio, because the workability of concrete increases with increase in maximum size of the aggregate. However, the compressive strength tends to increase

with the decrease in size of aggregate. Thus, a fine balance needs to be maintained to achieve a concrete of desired strength and workability.

- **Grading and type of aggregate**

  The grading of aggregate influences the mix proportions for a specified workability and water-cement ratio. Coarser the grading leaner will be the mix which can be achieved. Very lean mix is not desirable since it does not contain enough finer material to make the concrete cohesive. The type of aggregate influences strongly the aggregate-cement ratio for the desired workability and stipulated water cement ratio. An important feature of a satisfactory aggregate is the uniformity of the grading which can be achieved by mixing different size fractions.

- **Quality control**

  In actual field applications, concrete may exhibit deviation from the required compressive strength may occur due to the variations in the properties of the mix ingredients and lack of accuracy in batching, mixing, placing, curing and testing. The factor controlling this difference is termed as quality control. It primarily depends on the surrounding environmental conditions, the type of storage facilities available at the site for aggregates and cement. The degree of control required is generally estimated statistically by the variations in test results of representative cube or cylindrical concrete specimens.

1.9.4 Factors to be considered for mix design

The following factors form the primary inputs for a mix design:

- The grade designation giving the characteristic strength requirement of concrete.
- The type of cement influences the rate of development of compressive strength of concrete.
- Maximum nominal size of aggregates to be used in concrete may be as large as possible within the limits prescribed by IS 456:2000.
- The cement content is to be limited from the consideration of shrinkage, cracking and creep.
- The workability of concrete for satisfactory placing and compaction is related to the size and shape of section, quantity and spacing of reinforcement and technique used for transportation, placing and compaction.

  In the present study, the data generated from strength experiments carried out under the controlled laboratory condition is analyzed and used for development of models for prediction of concrete compressive strength.

1.10 Aim and Scope of the present study

Wide literature is available on the prediction of compressive strength of concrete. A variety of mathematical models including regression analysis techniques have been traditionally used to describe and predict the strength and other physical properties of cement based materials. Although the work of applying neural networks to prediction of concrete strength has been in use for more than two decades, but not much work has been done on applying these techniques to experimental data generated under controlled laboratory conditions. The literature, as reviewed, presents techniques, which consist mainly of semi empirical expressions derived from the analysis of limited experimental data, are based on simplified assumptions. Furthermore, these methods often lack the ability to either account for the effect of all the parameters involved individually or the combined effect of all variables. So, there is enough scope to work on the application of neural networks to the data specially obtained under controlled conditions in the laboratory involving variations in various concrete making parameters.

1.11 Organization of the thesis

Keeping these objectives in view, the research work has been carried out. The thesis on this work contains six chapters. Chapter wise summary of the thesis is given below.

Chapter - 1 presents an introduction to neuro-computing paradigm with special emphasis on prediction of compressive strength of concrete. There are a number of internal network parameters related to network architecture and training process that need to be adjusted on experimental basis to optimize the network performance. A brief description of the major internal parameters and training algorithms used in this thesis is given in this chapter.

Chapter - 2 on literature review presents the work done by various researchers in the field of prediction of compressive strength of concrete using regression analysis, neural network and GP modeling. This chapter also contains a review of the literature relating to the innovations in the field of neuro-computing paradigms.

Chapter - 3 outlines the investigations on the experimentally generated data for prediction of compressive strength of concrete. Data for the present work has been taken from the experiments conducted by Kumar, 2003. For generating a reliable data bank on concrete compressive strength, Kumar (2003) considered five parameters, namely, water-cementitious material ratio, cementitious content, water content, workability, and curing

ages in his experiments. The experiments were performed in controlled laboratory conditions. A set of 15 cubes for each of the mixes so proportioned were cast and tested after 28, 56 and 91 days of curing. Thus, an extensive data bank for analyzing the compressive strength of concrete had been generated and the same has been used in the present work. Factor analysis has been performed on the experimental data in order to decide variables for predicting the compressive strength of concrete with the help of SPSS. The explanation of factor analysis is presented in this chapter.

Chapter - 4 comprises the development of regression models for predicting compressive strength of concrete at different curing ages. Different regression models are developed and in order to deal with the problem of multi-collinearity.

Chapter - 5 deals with a neural network model that has been proposed to predict compressive strength of concrete. In the development of this model, several variants of training algorithms are experimentally investigated for proposing an efficient network model. Also, different architectural parameters such as the number of hidden neurons, learning rate, activation functions, performance goal, epochs for the fine tuning of neural network were empirically investigated. This chapter also presents the development of GP model for prediction of compressive strength of concrete at three different ages of curing *i.e.,* 28, 56 and 91 days. The comparison of proposed neural network model with the regression model (traditional model) and with the GP model (looming model) is also presented in this chapter.

Chapter - 6 presents the inferences drawn as a result of the various experiments conducted in this thesis. Also, some pointers to the future research on the topic under consideration in this thesis are discussed briefly.

CHAPTER 2

REVIEW OF LITERATURE

2.1 General

Literature review has been divided into two parts in this thesis. First part elaborates various recent applications of neural networks, regression analysis and genetic programming especially in the area of civil engineering; and the second part deals with the innovations in the field of design and development of neural networks.

2.2 Recent applications for prediction of compressive strength of concrete

Many researchers have studied and suggested various techniques for predicting the compressive strength of concrete. Some of these techniques are summarised below, in brief.

2.2.1 Prediction of compressive strength of concrete using regression models

Namyong *et al.* (2004) carried out multiple regression analysis for the prediction of compressive strength of concrete using water-cement ratio, cement contents and cement-aggregate ratio. They suggested two regression equations. First equation is used to predict the compressive strength of concrete at 7 days. This equation is:

$$f_7 = \exp\left[2.393 - 1.217\left({}^{w}/_{c}\right) - 0.0048c + 6.16\left\{{}^{c}/_{(s+g)}\right\}\right] \quad (2.1)$$

and the second equation to predict the compressive strength at 28 days is:

$$f_{28} = \exp\left[2.98 - 1.588\left({}^{w}/_{c}\right) - 0.00642c + 7.6888\{d(s+g)\}\right] \quad (2.2)$$

where, f_7 and f_{28} are prediction compressive strength at the age of 7 and 28 days respectively, *w/c* is water-cement ratio, *c* is cement contents, *w* is water contents, *d*(s+g) is cement-aggregate ratio.

Rajamane *et al.* (2007) predicted the compressive strength of concrete when sand is replaced by Fly Ash (FA). They suggested a formula to predict compressive strength of concrete at 28 days. They also performed a comparison of predicted and actual strength. The predicting equation is based on modified Bolomey equation. The equation also considered different levels of replacement of sand and it is also possible to account for cases when the quantity of FA added is more than that of sand replaced on weight basis. The suggested formula can be used to modify basic cement concrete mix so that the concretes with and without sand replacement by FA have similar strength.

Noorzaei *et al.* (2007) also developed neural networks for predicting compressive strength of concrete. Their study suggested that the use of neural networks has several significant advantages over other conventional methods. They found that the performance of 6-12-6-1 architecture was better than other architectures and the MSE for the training set was 5.33% for 400 training data points, 6.13% for the 100 verification data points and 6.02% for 139 testing data points. They concluded that neural networks could predict compressive strength of concrete with a confidence level of about 95%.

Zain *et al.* (2009) proposed a new mathematical model developed using non-linear regression equation for the prediction of concrete compressive strength at different ages. They developed two regression equations for prediction of compressive strength of concrete, first equation was to predict the compressive strength at 7 days. This equation is:

$$f_7 = a_0 c^{a1}.w^{a2}.fa^{a3}.ca^{a4}.\rho^{a5}.w/c^{a6} \quad (2.3)$$

and second equation was to predict the compressive strength at the age of 28 days:

$$f_{28} = a_0 c^{a1}.w^{a2}.fa^{a3}.ca^{a4}.\rho^{a5}.w/c^{a6} \quad (2.4)$$

where, f_7 and f_{28} are compressive strengths of concrete at 7 and 28 days, respectively. Here, *c* is cement, *w* is water, *fa* is fine aggregate, *ca* is coarse aggregates and *w/c* is water-cement ratio.

They concluded that these models could be used with any data set for prediction of compressive strength of concrete.

Wu *et al.* (2010) developed a regression model for the prediction of compressive strength of concrete. They also performed principal component analysis and identified eight process variables, *i.e.,* cement, water, furnace slag, FA, fine aggregate and curing age. They reported that their model has 84.37% coefficient of determination (R^2), which can statistically be considered as a good fit.

Chou *et al.* (2011) performed quantitative analysis by using five different data mining methods. Out of these five methods, two were machine learning models (neural network and Support Vector Machine), one was statistical model (multiple regression) and other two were meta-classifier models (multiple additive regression trees and bagging regression trees). They tested developed models with a dataset taken from 17 concrete mix designs developed at laboratories. They revealed that the multiple additive regression tree (MART) was superior in prediction accuracy, training time and ad-version to over fitting trends. They also concluded that according to analytical results, MART-based modeling is effective for predicting the compressive strength of varying high performance concrete ages.

Hasan and Kabir (2012) developed a simple mathematical model to predict compressive strength of concrete at the age of 28 days from early age results collected experimentally. A polynomial surface equation was developed in terms of two constants and concrete strength of a particular day. They concluded that their equations have a potential to predict concrete strength for any curing age with high accuracy.

Sayed-Ahmed (2012) developed a statistical model to predict compressive strength of concrete containing different mixtures at any fixed age or at different ages of 1, 3, 7, 28, 56, 91 and 180 days. He reported that his study addressed the effect of different mixtures of concrete on the compressive strength and this information would help the cement industry in producing the required concrete strength. He concluded that his model could provide high correlation to experimental results for compressive strength of concrete.

Chore and Shaelke (2013) carried out multiple regression analysis for prediction of compressive strength of concrete. They argued that their models could yield good correlation coefficient for prediction of compressive strength at the ages of 7, 28 and 40 days. They concluded that derived formulas provided an effective and straightforward analysis tool.

2.2.2 Prediction of concrete compressive strength using neural network models

Ghaboussi *et al*. (1991) investigated the use of neural networks to model the stress-strain behaviour of plain concrete subjected to different loading conditions.

Goh (1995), Mukherjee and Biswas (1997), Yeh (1998), Sanad and Saka (2001) and Lee (2003) studied the feasibility of using neural networks to predict compressive strength of concrete under different conditions.

Glass *et al.* (1997) investigated the use of neural networks to model binding in concrete. Buenfeld and Hassanein (1998) and Haj-Ali *et al.* (2001) proposed different neural network models to predict durability of concrete subjected to various degradation mechanisms.

Hong-guang and Ji-zong (2000) suggested a method to predict 28 days compressive strength of concrete by neural networks. They proposed multilayer feedforward neural networks based on inadequacy of present methods dealing with multiple variables and non-linear problems. For this purpose, they considered 14 test parameters that could influence compressive strength of concrete. The neural network architecture used by them is 11-7-1.The results of their study showed that neural network models have higher prediction accuracy and the use of proposed model could be very beneficial to concrete industry.

Lee (2003) developed I-PreConS (Intelligent PREdiction system of CONcrete Strength) which could provide in-place strength information of concrete. To achieve this purpose, he developed five multiple architectures for day 2 to day 28. He has reported test results for all test patterns well above an accuracy of 90.00%.

Akkurt *et al.* (2004) suggested a fuzzy logic model for prediction of compressive strength of concrete. They also compared this model with the neural network model to ensure its error levels and ease of application. As per their study, fuzzy model yielded slightly higher error than neural networks. They also reported that fuzzy logic algorithm is more user-friendly and more explicit model; and neural networks can predict compressive strength of concrete within low error margins successfully.

Kenealramani and Gupta (2006) conducted prediction of compressive strength of concrete using ultra sonic pulse velocity through neural networks and multiple regression analysis. They also performed comparison between these two models. The results of their study presented a good degree of coherence for both shapes and sizes of concrete specimens.

Oztas *et al.* (2006) predicted compressive strength and slump of High Strength Concrete (HSC) using neural network. They aimed to show possible applicability of neural networks to predict compressive strength and slump of HSC. They developed a neural network model, trained it and tested it using available test data of 187 different concrete mix designs gathered from literature. The data used in neural network model were arranged in a format of seven input parameters and two output parameters *i.e.,* compressive strength and slump value. They reported R^2 of test data set for compressive strength of concrete and slump values of concrete as 99.93% and 99.34%, respectively. They concluded that proposed neural network model has strong potential as a feasible tool for predicting compressive strength and slump values of concrete.

Nataraja *et al.* (2006) developed a fuzzy-neuro model for normal concrete mix design. In their study, they also presented development of a novel technique for approximate proportioning of standard concrete mixes. They developed distinct fuzzy inference modules with five layers and framed to capture vagueness and approximations in various steps of design as suggested in IS: 10262-2003 and IS: 456-2000. They concluded that methods proposed in their study are good for various grades of standard concrete mixes than conventional methods.

Pala *et al.* (2007) developed neural networks to predict long term effects of FA and Silica Fume (SF) on compressive strength of concrete cured for a long period of time. Their investigation also covered concrete mixes at different water cementitious material ratios. These concrete mixes also

contained low and high volumes of FA and with or without additional small amount of SF. They gathered 24 different mixes with 144 different samples from literature for this purpose. These samples were cured at the ages of 3, 7, 28, 56 and 180 days. A multilayer neural network model was developed, trained and tested using this data set. The data used for neural network model were arranged in a format of eight input parameters and one output parameter, which is compressive strength of concrete. The results of their study presented neural network model as a feasible and strong tool for evaluation of effect of cementitious material on the compressive strength of concrete. They reported R^2 for training and testing data sets as 99.85% and 99.90%, respectively.

Tapkin *et al.* (2007) estimated compressive strength of concrete using ultra sonic pulse velocities and neural networks. They used multilayer feed-forward back propagation neural network model and compared prediction performances with RMSE values. They also compared this model with other studies in literature and concluded that RMSE values were reasonably small and concluded that estimates were fairly good.

Topcu and Saridemir (2008) developed a model for predicting compressive strength of concrete containing FA using neural networks and fuzzy logic. This model could predict for curing ages of 7, 28 and 90 days of compressive strength of concrete. They gathered 52 different mixes with 180 specimens from literature. They arranged data for neural networks and fuzzy logic models in a format of 9 parameters and an output parameter which is the compressive strength of concrete. They reported R^2 for test data set in neural networks and fuzzy model as 99.90% and 99.72%, respectively.

Fazel Zarandi *et al.* (2008) developed Fuzzy Polynomial Neural Network (FPNN) for approximation of compressive strength of concrete and verified applicability of FPNN using six models. In their study, they used combination of fuzzy neural networks and polynomial neural network. They used Root Means Square Error (RMSE) and Correlation (R) for the evaluation and comparison of the models. They revealed that FPNN-Type1 has strong potential and could be used as a feasible tool for prediction of compressive strength of concrete.

Ozturan *et al.* (2008) compared regression techniques for compressive strength of concrete with neural network approach. In their paper, they aimed to illustrate that neural network can be used for prediction of low to medium strength concretes at the curing age of 28 days. They also compared the accuracies of neural networks and multiple regression models on the basis of R^2. They concluded that encouraging results are obtained by neural network models using data for fresh concrete and early strength simultaneously.

Karahan *et al.* (2008) devised neural network models for strength properties of Steel Fiber Reinforced Concrete (SFRC). The specimens were cured under the standard conditions for 7, 28, 90, 365 days. They used different training algorithms, namely, Levenberg- Marquardt (LM), Scaled Conjugate gradient (SCG) and Fletcher-Powell Conjugate gradient (CGF) algorithms. They concluded that LM algorithm is the best suitable algorithm for prediction of long-term strength of SFRC containing FA.

Alshihri *et al.* (2009) developed neural networks for predicting compressive strength of structural Light Weight Concrete (LWC). In their investigation, neural networks were used to predict compressive strength of LWC mixtures at the curing ages of 3, 7, 14 and 28 days. They used two models. Out of these two, one was feed-forward Back Propagation (BP) and other was Cascade Correlation (CC) neural network. They concluded that CC neural network model predicted slightly more accurate results and learned very quickly as compared to BP model. The findings of their study indicated that neural network models can be used for estimating compressive strength of LWC. They also reported that CC model can reduce cost and save time particularly in such class of problems and could help to avoid conducting costly experimental tests that require specialized equipments and human expertise.

Raghu Prasad *et al.* (2009) developed neural network models to predict compressive strength of Self Compacting Concrete (SCC) and High Performance Concrete (HPC) with high volume FA. They gathered data from literature. In their study, they showed that high volume FA in SCC and HPC affects a good increase in compressive strength at the curing ages of 56 days and 90 days, but not so much at the curing age of 28 days. They also reported that addition of 3.4% micro-silica, with high volume FA in SCC or HPC for target strength of 28 days, was sufficiently high when considered at the curing age of 90 days. They reported that network presented in their work can predict compressive strengths over a wide range of compressive strengths of concrete from about 30 to 60 MPa.

Rasa *et al.* (2009) developed neural network models to predict density and compressive strength of concrete containing SF. They considered compressive strength at the curing age of 28 days and Saturated Surface Dry (SSD) density values as an output parameters of prediction. This study indicated that density and compressive strength of concrete can be predicted much more accurately using neural network models as compared to existing conventional methods.

Basyigit *et al*. (2009) developed neural network and FL models to predict compressive strength of heavy weight concrete. In their study, they used two different types of aggregates. Out of these two,

one was normal and second was barite aggregate. They investigated compressive strength of all mixtures at the curing ages of 7, 28 and 90 days. They concluded that compressive strength of concrete can be estimated using their neural network models and FL model without performing any other experiments.

Jamil *et al.* (2009) developed neural network models for optimization of HPC mix design incorporating SF, FA and rice husk ash. They reported that suggested neural network models provide an efficient and rapid method of obtaining optimal solutions and demonstrated its ability very well in training for given input/output patterns. They concluded that application of neural network in the field of HPC mix design is very appropriate in order to preserve and disseminate valuable experience and innovation efficiently at reasonable cost.

Ozcan *et al.* (2009) performed comparison of neural network and FL models for prediction of long term compressive strength of SF concrete. They produced a dataset of a total 48 concrete mixes in the laboratory conditions and utilized this set for neural network and FL models. Their neural network model predicted compressive strength of SF concrete with R^2 of 99.44% and 97.67% for training and testing, respectively. The FL model predicted compressive strength of SF concrete with R^2 value 92.74%. They concluded that neural network and fuzzy models can be used as alternative approaches for the prediction of compressive strength of SF concrete.

Erdal (2009) predicted compressive strength of vacuum processed concretes using neural network models and regression techniques. In his study, he developed regression equations and neural network models for estimation of compressive strength of vacuum processed concrete. He also compared prediction performance of earlier regression equations developed by him with his neural network model. He concluded that R^2 achieved in training and validation of data for neural network was 91.13% which was better than R^2 of regression equations used.

Saridemir (2009) developed a model to predict compressive strength of concrete containing metakaolin and SF by neural networks at the curing ages of 1, 3, 7, 28, 56, 90 and 180 days. They gathered data of 195 specimens produced with 33 different mixture proportions from literature. They arranged data in a format of eight input parameters for neural network model. They concluded that neural networks have strong potential for predicting 1, 3, 7, 28, 56, 90 and 180 days compressive strength values of concretes containing metakaloin and SF.

Alilou and Teshnehlab (2010) developed neural network models for prediction of compressive strength of concrete. They collected data by conducting experiments on concrete samples in the laboratory. They produced 432 specimens. Out of 432 specimens, 300 data samples were used to

train neural network, 66 data samples were used to validate and 66 data samples were used to test the model. They suggested that their model can be used as a very useful tool for reducing duration of project execution and a precise prediction of 28 days compressive strength of the concrete can be obtained on day 3 using their suggested model.

Hakim *et al.* (2011) proposed a method to predict 28 days compressive strength of high strength concrete (HSC) by using multiple feed-forward neural networks. They developed, trained and tested model using a total of 368 different data of HSC mix designs collected from literature. They studied a total of 30 architectures and reported that 8-10-6-1 architecture was the best architecture for HSC. Their results showed that relative percentage error (RPE) for training set was 7.02% and for testing set, it was 12.64%. They also concluded that neural network models have high prediction accuracy.

Muthupriya *et al.* (2011) developed neural network models for predicting compressive strength of cubes and durability of concrete containing metakaolin with FA, and SF with FA at the curing ages of 3, 7, 28, 56, and 90 days. They used 140 specimens produced with 7 different mixture proportions and designed data in a format of 8 input parameters for neural network models. They reported that neural networks have high potential to predict compressive strength of concrete and durability values of concrete containing metakaolin, SF and FA.

Aggarwal and Aggarwal (2011) presented potential of FL models and neural network models for prediction of compressive strength of concrete mixtures. They collected data from literature and observed that their developed FL model can adjust itself to any type of linear or non-linear form through fuzzy subsets of linguistic compressive strength variables. They concluded that FL models performed better than neural network models.

Uysal and Tanyildizi (2011) carried out a study to predict core compressive strength of SCC mixtures with mineral additives. They reported that they used different neural network algorithms and reported Fletcher-Powell Conjugate algorithm is the best learning algorithm to predict core compressive strength of SCC with high correlation coefficient.

Diab *et al.* (2012) divided his work in two phases. One of the phases was to include validation of neural networks to predict mortar and concrete properties due to sulphate attack and second phase was related with design charts. They reported that neural network model has high efficiency for predicting compressive strength of concrete, expansion and weight loss due to sulphate attack.

Abdollahzade *et al.* (2011) developed BP neural networks to predict compressive strength of rubberized concrete based on test measurements. They concluded that their model has ability to

predict strength with acceptable degree of accuracy in comparison to multiple linear regression model.

Madandoust *et al.* (2012) developed Generalized Group Method of Data Handling (GMDH)-type neural network and Adaptive Neuro-Fuzzy Inference Systems (ANFIS) based models for predicting standard cube strength of concrete based on core testing which was influenced by a number of factors. They concluded that results obtained from the GMDH-type neural networks and ANFIS models have a high degree of coherence with experimental results. They reported that prediction of compressive strength of concrete based on core test results can be carried out quite accurately using their suggested model.

Wankhade and Kambekar (2013) developed data mining approach to predict compressive strength of concrete and assessed its prediction reliability for HPC. They compared proposed approaches using performance measures, *i.e.,* correlation (*R*), Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) to obtain a widespread comparison of applied techniques. They observed that standard error back propagation performed very well in terms of higher *R* and lower RMSE. They reported that neural networks can be looked upon as an alternative approach in prediction of compressive strength of concrete.

Dantas *et al.* (2013) developed neural network models for prediction of compressive strength of concrete containing construction and demolition waste. They used extensive data set of 1178 specimens from literature. They reported that neural network models provided a good accuracy of prediction at the curing ages of 3, 7, 28, and 91 days compressive strength of concrete.

Khan *et al.* (2013) used multi-layer feed-forward neural network models to predict compressive strength of concrete. They reported that neural network models estimated compressive strength of concrete much closer to experimental results as compared to existing mathematical models.

Akande *et al.* (2014) studied the performance of SVM for compressive strength of concrete and also compared results of SVM with neural networks. They concluded that SVM displayed slightly a better performance as compared to neural networks and is also highly stable.

Cheng *et al*. (2014) developed the Genetic Weighted Pyramid Operation Tree (GWPOT) model to solve problem of predicting HPC compressive strength. They compared results of their study with other prediction techniques including neural networks, SVM and Evolutionary Support Inference Model (ESIM). They concluded that proposed GWPOT model had generated explicit formulas, which provide important advantages in practical applications.

Tiryaki and Aydin (2014) designed neural network models to predict compressive strength parallel to grain of heat treated woods, without doing any other compressive experiments. They also compared neural network models with multiple linear regression and their results indicated that neural networks provided a better prediction as compared to multiple regression model.

Chou *et al.* (2014) performed a comprehensive comparison of various learning techniques used individually and in combination for performing simulations for compressive strength of concrete based on dataset collected from several countries. They used different machine learning techniques. These techniques are multilayer perceptron, SVM, classification and regression tree and linear regression supported by voting, bagging and stacking approaches. These approaches were used to combine multiple single learning based models. They validated applicability of machine learning, voting, bagging and stacking techniques using simple and efficient simulations of compressive strength of concrete.

Nikoo *et al.* (2015) developed neural network models and used 173 experimental samples of cylindrical concrete parts with different characteristics and compressive strengths of concrete. They also optimized number of layers, nodes and weights using Genetic Algorithm (GA). To evaluate accuracy of model, they also compared their optimized neural network model with multiple linear regression model. They recommended that neural network model showed more accuracy, flexibility, capability and accuracy in predicting compressive strength of concrete.

Moutassem *et al.* (2015) reviewed theoretical and phenomenological model for predicting compressive strength of concrete present in literature. They investigated applicability and accuracy of their models for prediction of compressive strength of concrete at 28 days and at different ages using the experimental data available in literature. They concluded that majority of models give acceptable predictions because strength is affected more by water-to-cement ratio as compared to aggregate properties and gradation.

2.2.2.1 Statistical measures for testing performance of neural network models

The literature reveals that commonly used measures for evaluating the performance of neural network models include the Root Mean Square Error (RMSE) (Altendorf *et al.* 1999; Salehi *et al.* 1998a; Grzesiak *et al.* 2003), sensitivity analysis, cross-validation (Coakley, 1991; Kwok and Yeung, 1995; Drummond *et al.* 2003), and Relative (or Receiver) Operating Characteristics (ROC) curve (Brethour, 1994; Yang *et al.* 1999; Pietersma *et al.* 2003), coefficient of determination R^2 (Altendorf *et al.* 1999; Drummond *et al.* 2003; Erenturk *et al.* 2004). Several techniques for

improvement of the accuracy of prediction and the robustness of neural network models during the training and final tuning of the neural network have also been described in the literature (Fang *et al.* 1998).

2.2.3 Prediction of compressive strength of concrete using genetic programming models

Baykasoglu *et al.* (2004) predicted compressive strength of concrete strength using GP and neural networks. They also performed stepwise regression analysis to have an idea about predicted power of soft computing techniques. The R^2 obtained was 77.50%, 69.60%, 35.70% for GP, neural networks and regression models, respectively. They concluded that stepwise regression analysis does not performed better as compared to GP and neural network models.

Thamma and Barai (2009) developed GP model to predict 28 days compressive strength of concrete and compared their model with FL model. The results showed that GP model performed better than FL model.

Saridemir (2010) developed two GP models for predicting compressive strength of concrete containing rice husk ash at the curing ages of 1, 3, 7, 14, 28, 56 and 90 days. He suggested that GP is a strong method for prediction of compressive strength of concrete containing rice husk ash.

Heshmati *et al.* (2010) proposed novel model formulations of compressive strength and slump flow of HPC mixes. They used dataset from previously published literature and also compared their results of suggested formulations with other existing models and formulas found in literature. They reported that their suggested formulas are able to predict strength and slump flow to a higher degree of accuracy.

Saridemir (2010) developed a model for predicting compressive strength of concrete containing rice husk ash at the curing ages of 1, 3, 7, 14, 28, 56 and 90 days. They used a dataset of 188 specimens (produced with 41 different mixture proportions) from literature. They compared results of training, testing and validation sets of the models proposed by them. They reported that GP is a good soft computing technique for use in prediction of compressive strength of concrete and it may open a new area of accurate and effective formulation of many civil engineering applications.

Dadkhah and Esfahani (2013) explored two soft computing techniques: ANFIS and GP. They used a dataset of 130 tuples. They concluded that ANFIS and GP models are capable of predicting uni-axial compressive strength of concrete and they also reported that GP has better results of prediction as compared to ANFIS.

Gorphade *et al.* (2014) developed application of GP for predicting workability and strength of HPC. They used 300 examples to train model and weights were obtained using a GP approach. They reported that an accuracy of about 95.00% and concluded that their neural network model could serve as a macro-mechanical model for predicting workability and strength of HPC.

2.3 Design and development of neural network models

The choice of an optimal network design for a given application still remains an empirical art (Fogel, 1991; Kecman, 2001; Freeman and Skapura, 2004). However, it is always possible that by giving sufficient layers and neurons, more than one network architecture can be trained to solve a specific problem to a given degree of error provided sufficient time is given. The designer is then left with the decision as to which network is best, *i.e.*, which network will sufficiently achieve the user's purpose when placed in actual practice. The network architecture is a very important consideration for the optimal trainability and generalisation ability (Yu *et al.* 1993; Bebis and Georgiopoulos, 1994). Several researchers have rigorously analysed the properties of neural networks having diverse designs including feed-forward connections, a variety of nonlinear processing neurons, and several suitable training procedures, *etc*. A brief description to this effect is given below.

i. Optimum number of hidden layers

Lippmann (1987) and Hornik *et al.* (1989) have shown that a three-layer perceptron (*i.e.*, a multilayer feedforward neural network) with sigmoid-type nonlinearity at cells can approximate any arbitrary nonlinear function and generate any complex design region needed for classification and recognition tasks; whereas Chester (1990) and Sontag (1992) argued that two hidden layers are sufficient to achieve full generality.

Mukherjee and Deshpande (1995), Desai and Bharati (1998) revealed that there exists no general method for determining the optimum number of hidden layers and number of neurons in each hidden layer so as to find an appropriate model. We cannot generalise the number of hidden layers and neurons in each hidden layer probably because they are a function of the expected intelligence.

Choi *et al*. (2006) developed a separate learning algorithm which includes a deterministic and heuristic approach. In this algorithm, input nodes to hidden nodes and hidden nodes to output nodes are separately trained. It solved the local minima in two layered feed forward network. They argued that they achieved the best convergence speed.

Panchal *et al.* (2011) proposed a methodology to analyze the behaviour of multi-layer perceptron. They concluded that number of hidden layers is inversely proportional to the minimal error.

Neural networks with more than two hidden layers are found very rare in the civil engineering and allied applications.

ii. Optimum number of neurons in each hidden layer

Reed (1993) and Murata *et al.* (1994) reported that number of hidden neurons affects the network performance. They concluded that if the number of hidden neurons is less, then network can perform better.

Fang *et al.* (2000) and Kinser (2000) concluded in their respective studies that networks with less number of hidden neurons sometimes do not converge during training.

Kominakis *et al.* (2002) and Ghedira and Bernier (2004) reported a simple formula for number of hidden neurons: the product of 'number of patterns used for training' and 'the number of input variables' is used as number of neurons in the first hidden layer whereas number of neurons for second hidden layer is considered as twice the number of patterns used for training.

Yuan *et al.* (2003) concluded that best number of hidden neurons of a network can be estimated by firstly constructing a decision tree based information gains of all partitions among hidden neurons' activation values, and then searching decision rule of tree's nodes to find out important hidden neurons.

Doan and Liong (2004) provided a formula for calculation of number of hidden neurons. They proposed number of neurons for hidden layer can be determined by $m = \text{interger}\left(n/2 + \sqrt{P}\right)$, where n is the number of input neurons and P is the dataset size.

Jinchuan and Xinzhe (2008) investigated a formula and tested it on 40 cases. The formula is

$$N_n = (N_{in} + \sqrt{N_P})/L \tag{2.5}$$

where L is the number of hidden layers, N_{in} is the number of input neurons and N_p is the number of input samples. They had reported that number of hidden layers and hidden units depend on complexity of the network architecture, number of input and output units, number of training samples, degree of the noise in the sample data set and training algorithm used.

Yuan *et al.* (2010) proposed a method of estimation of hidden neuron based on information entropy. The goal had been to avoid over learning problem because of exceeding number of hidden neurons and to avoid shortage of capacity because of few hidden neurons.

However, most statisticians are convinced that rules of thumbs are not of any use. They argued that there is no way to determine a good network topology from just number of inputs and outputs (Sarle, 1997). As such, a general approach is yet to emerge for deciding optimum number of neurons in each hidden layer.

iii. Learning parameters

Dayhoff (1990) has argued that some experiments indicated that if learning rates are allowed to decrease during training, then algorithms can sometimes produce better learning. He also pointed out that very large learning rates can lead to unsatisfactory learning and unstable networks, while small values result in slow learning. Since the purpose of training a neural network is to minimise global prediction error of dataset, it is important to understand the effects of different internal parameters involved in network training. He suggested that these effects may be worthy of investigation, if there is lack of information, along with the existence of conflicts concerning some parameter values.

Dowell (1993) used different combinations of learning parameters, *viz.*, learning rate, momentum, number of neurons, and number of learning cycles (epochs), to improve accuracy of measuring peanut quality. The study revealed that learning rate alone had no significant effect on the classification of damaged and undamaged peanut kernels. However, the results varied for different combinations of the learning rate, number of neurons, numbers of epochs, and momentum.

Freeman (1993) comprehended that value of the learning rate affects network performance. As a result, it was suggested that an increase in this value, as learning proceeds and the network error decreases, could accelerate convergence, but also, if the value were too large the network would lose its stability. They specified that range of learning rates for their study was varied between 0.05 and 0.25.

In majority of works reported in the literature, a fixed-value of learning rate is used during training of the neural networks (Liu *et al.*, 2001). However, online adjustable (or adaptive) learning rate facilitates network to perform better (Ferentinos and Albright, 2002; Benedettia *et al.*, 2004; Hessami *et al.*, 2004; Ferentinos, 2005).

Although neural network structures and configurations are application-dependent, results from similar applications and data structures may provide specific methods and parameter values that can be generalised (Salehi *et al.*, 1998b). The study also suggested that the users of neural

networks should pay attention to values of different learning parameters and to the method of data presentation. They should also carry out several repetitions, using different initial weight values, in order to optimise network results.

2.4 Gaps in the literature

The previous sections present a detailed review of the current literature on the existing mathematical models for prediction of compressive strength in concrete. It is very clear from Section 2.2 that wide literature is available for prediction of compressive strength of concrete. A variety of mathematical models including regression analysis have been traditionally used to describe and predict compressive strength of concrete and other physical properties of cement based materials. Section 2.3 presents a review of the possible parameters of neural network design. It is evident from review of the literature that there exist no precise criteria to decide the various network internal parameters as listed in Section 2.3 to develop an application-specific neural network model. The following gaps have been identified in the existing literature:

- Although the work of applying neural networks to prediction of concrete strength has been in use for more than two decades, not much work has been done on applying these techniques to experimental data generated under controlled laboratory conditions.
- The currently existing methods of prediction of compressive strength often lack the ability to either account for the effect of all the parameters involved individually or the combined effect of all variables.

Considering the gaps as above, the following objectives had been proposed for this research work:

- To apply factor analysis on the experimental data in order to decide variables for predicting the compressive strength of concrete.
- To develop an efficient neural network model for predicting compressive strength of concrete.
- To compare the developed neural network model with the existing conventional methods.

CHAPTER 3

EXPERIMENTAL DATASET AND ITS FACTOR ANALYSIS

3.1 Introduction

The data published by Kumar (2003) has been considered for different experiments in this work. For generating a reliable data bank on compressive strength of concrete, he considered five parameters, namely, water-cementitious material ratio (*w*/*cm*), cement content (*c*), water content (*w*), workability and curing ages in his experiments. The data was collected in controlled laboratory conditions. Data analysis in a research activity plays an important role. Subsequent interpretation of data and its responses are very critical to the final outcome. Therefore, analysis of the dataset given by Kumar (2003) has been done with the help of Statistical Package for the Social Sciences (SPSS). We have also carried out factor analysis in order to decide efficient variables for prediction. As such, this chapter discusses dataset used, procedure followed for the analysis of dataset and this also highlights variables that have highest influence on compressive strength of concrete.

3.2 Details of the dataset used

Kumar (2003) generated the dataset using different values of parameters for concrete mixes. The materials used in these mixes are briefly described below.

3.2.1 Materials used

Cement (*c*)

Although all materials that go into the concrete mix are essential, cement is the most important component as it is usually the most delicate link in the chain. Primarily, the function of cement is to bind the fine aggregates (*fa*) and coarse aggregates (*ca*) together and also to fill the void between fine aggregate and coarse aggregate particles resulting in formation of a compact mass. Although, it constitutes only about 20% of the volume of concrete mix, it is the active portion of binding medium and is the only scientifically controlled ingredient of concrete. Any variation in its quantity affects compressive strength of the concrete mix.

Ordinary Portland cement of 43 grade conforming to IS: 8112-1989 has been used for generating the dataset for compressive strength of concrete. The value of cement content has

been varied from 350 kg/m^3 to 475 kg/m^3 and mixes have been proportioned corresponding to cement contents of 350, 375, 400, 425, 450 and 475 kg/m^3.

Table 3.1: Range of values of various parameters

Water-cementitious ratio (*w*/*cm*)	0.42 – 0.55
Cement content (*c*)	350 – 475 @ 25 kg/m^3
Water content (*w*)	180-230 @ 10 kg/m^3
Workability	Medium and high
Curing ages, days	28, 56, 91
Percentage replacement cement by fly ash (FA)	0 to 15 per cent

Aggregates

The aggregate is the matrix or principal structure consisting of relatively inert and coarse particles. IS: 383-1970 defines fine aggregate (*fa*) as the one passing 4.75mm IS sieve. The fine aggregate is often termed as a sand size aggregate. This standard further defines coarse aggregate (*ca*) as the one that are retained on 4.75 mm IS sieve. The coarse aggregate is primarily used for the purpose of providing bulk to the concrete. The most important function of fine aggregate is to assist in producing a workable and a uniform concrete mix. The fine aggregate, in addition, also assists the cement paste to hold coarse aggregate particles in suspension. The aggregates provide about 75% of the body of concrete and hence their influence is extremely important. The properties of these particles greatly affect the performance of concrete.

To increase the density of the resulting concrete mix, coarse aggregate is used in two or more sizes. Three types of aggregate with different sizes have been used for generating the dataset. The details of the same are as below:

CA-I (*ca*1): Aggregate passing 20 mm sieve and retained on 10 mm sieve.

CA-II (*ca*2): Aggregate passing 10 mm sieve and retained on 4.75 mm sieve.

CA-III (*ca*3): Aggregate passing 4.75 mm sieve and retained on 2.36 mm sieve.

The percentage contributions from each type of aggregates have been varied and three zones have been considered for proportioning of the concrete mix. The details of three zones are given in Table 3.2. Table 3.3 provides the physical properties of all the materials used in development of concrete mixes.

Table 3.2: Zones of aggregates

Zone	Percentage passing 20mm sieve and retained on 10mm sieve	Percentage passing 10mm sieve and retained on 4.75mm sieve	Percentage passing 4.75mm sieve and retained on 2.36mm sieve	Fineness modulus
A	67	33	-	6.67
B	50	50	-	6.50
C	-	50	50	5.50

Table 3.3: Physical properties of materials

Materials	Properties
Ordinary Portland Cement (OPC)	Grade: 43, as per IS:8112-1989 Specific gravity: 3.12 7 days compressive strength: 35.50 MPa 28 days compressive strength: 46.50 MPa
Fine Aggregates (*fa*)	Zone: II Fineness modulus: 2.09 Specific gravity: 2.54
Coarse Aggregates - I (*ca*1) 20mm size	Specific gravity: 2.61
Coarse Aggregates - II (*ca*2) 10mm size	Specific gravity: 2.63
Course Aggregates - III (*ca*3) 4.75mm size	Specific gravity: 2.58

Water (*w*) content requirements for each zone of aggregate has been selected based upon the workability requirement. The variations in water content for each zone of aggregate are given in Table 3.4. The lower limits of water content in Table 3.4 correspond to medium workability and upper limits to high workability. Within these limits, mixes have been designed incrementing the value of *w* by 10kg/m^3 for each zone.

Table 3.4: Water requirements for each zone of aggregate

Zone	A	B	C
Water content (*w*), kg/m^3	180 - 210	190 - 220	200 - 230

Fly ash (FA) is a by-product of the combustion of pulverized coal in thermal power plants. It is removed by the dust collection system as a fine particle residue from the combustion gases before they are discharged into the atmosphere. The addition of FA in concrete mix affects all

aspects of concrete properties. As a part of the composite that forms the concrete mass, FA acts in part as fine aggregates and in part as cementitious component. In recent years, there has been a recognition that FA differs in significant and definable terms, reflecting their composition and, to some extent, their origin. Canadian and American specifications recognize two general classes of FA: Class - C, normally produced from lignite or sub bituminous coals; and Class - F, produced from bituminous coals. Class - C ashes differ from the Class - F ashes, principally, in often having a capacity for self-hardening in the absence of cement. Also Class - C ashes have high levels of calcium content (containing 15 to 35 percent CaO) as compared to Class - F ashes (containing less than 10 percent CaO). The FA used in present study is a low calcium Class - F FA and it conforms to the requirements of IS: 3812- 1981.

3.2.2 Compressive strength (f_c)

Compressive strength is the maximum compressive stress that, under a gradually applied load, a given solid material can sustain without fracture. The compressive strength of concrete is the most common performance measure used by engineers in designing buildings and other structures. The details of the proportions for concrete mixes without FA are shown in Table 3.5 and the compressive strength data at varying curing ages for these mixes is presented in Table 3.6. Table 3.7 shows proportions of ingredients of mixes containing 15% FA as substitution of cement and Table 3.8 gives the compressive strength data for all ages using FA in concrete mixes.

3.3 Factor Analysis

Factor analysis is a statistical technique used to depict variability among correlated and observed variables in terms of a potentially lower number of unobserved variables called factors. It is generally used for the data reduction purposes to get a small set of variables from a large set of variables. There are two types of factor analysis: exploratory factor analysis (EFA) and confirmatory factor analysis (CFA). EFA is used when there is no predefined idea about the structure or about the dimensions in a set of variables. CFA is used when there is a need to test specific hypothesis about the structure or number of dimensions underlying a certain set of variables. In the present study, the structure or dimensions of dataset is unknown, thus EFA is more steadfast method which facilitates to reduce large number of correlated variables to a small number of variables.

Table 3.5: Details of proportions for concrete mixes without FA

S. No.	Mix designation	*w/cm*	Mix proportions (*cm* : *fa* : *ca*)	Cement content (*c*) Kg/m^3
1	MD-1	0.533	1:1.58:3.05	375
2	MD-2	0.500	1:1.43:2.82	400
3	MD-3	0.525	1:1.54:2.99	400
4	MD-4	0.471	1:1.28:2.58	425
5	MD-5	0.494	1:1.39:2.77	425
6	MD-6	0.444	1:1.14:2.35	450
7	MD-7	0.467	1:1.25:2.54	450
8	MD-8	0.421	1:1.05:2.19	475
9	MD-9	0.442	1:1.19:2.46	475
10	MD-10	0.533	1:1.58:3.05	375
11	MD-11	0.500	1:1.43:2.82	400
12	MD-12	0.525	1:1.54:2.99	400
13	MD-13	0.471	1:1.28:2.58	425
14	MD-14	0.494	1:1.39:2.77	425
15	MD-15	0.518	1:1.51:2.95	425
16	MD-16	0.444	1:1.14:2.35	450
17	MD-17	0.467	1:1.25:2.54	450
18	MD-18	0.489	1:1.37:2.73	450
19	MD-19	0.421	1:1.05:2.19	475
20	MD-20	0.442	1:1.19:2.46	475
21	MD-21	0.463	1:1.23:2.51	475
22	MD-22	0.518	1:1.43:2.02	425
23	MD-23	0.489	1:1.29:1.86	450
24	MD-24	0.511	1:0.391:1.9	450
25	MD-25	0.463	1:1.17:1.72	475
26	MD-26	0.484	1:1.26:1.83	475
27	MD-27	0.514	1:1.39:3.26	350
28	MD-28	0.543	1:1.49:3.42	350
29	MD-29	0.475	1:1.25:2.99	375
30	MD-30	0.507	1:1.35:3.19	375
31	MD-31	0.450	1:1.10:2.70	400
32	MD-32	0.475	1:1.21:2.92	400
33	MD-33	0.423	1:0.98:2.47	425
34	MD-34	0.447	1:1.09:2.68	425
35	MD-35	0.422	1:0.98:2.45	450
36	MD-36	0.543	1:1.49:3.42	350
37	MD-37	0.507	1:1.35:3.19	375

38	MD-38	0.475	1:1.21:2.92	400
39	MD-39	0.447	1:1.09:2.68	425
40	MD-40	0.422	1:0.98:2.45	450
41	MD-41	0.533	1:1.47:2.41	375
42	MD-42	0.500	1:1.32:2.21	400
43	MD-43	0.525	1:1.44:2.36	400
44	MD-44	0.471	1:1.19:2.03	425
45	MD-45	0.494	1:1.29:2.18	425
46	MD-46	0.444	1:1.07:1.86	450
47	MD-47	0.467	1:1.17:2.00	450
48	MD-48	0.421	1:0.95:1.68	475
49	MD-49	0.442	1:1.06:1.84	475

Table 3.6: Details of compressive strength of concrete mixes without FA

S. No.	Mix designation	*w/cm*	f_{c28} (MPa)	f_{c56} (MPa)	f_{c91} (MPa)
1	MD-1	0.533	36.84	40.92	44.52
2	MD-2	0.500	43.13	50.22	51.97
3	MD-3	0.525	38.58	45.51	47.49
4	MD-4	0.471	47.16	51.25	54.27
5	MD-5	0.494	45.05	50.72	52.85
6	MD-6	0.444	49.63	54.48	58.04
7	MD-7	0.467	47.42	51.34	55.30
8	MD-8	0.421	54.01	57.91	60.15
9	MD-9	0.442	50.05	55.72	58.31
10	MD-10	0.533	37.81	43.50	47.55
11	MD-11	0.500	44.11	50.98	52.56
12	MD-12	0.525	40.90	46.56	51.07
13	MD-13	0.471	47.51	52.92	54.47
14	MD-14	0.494	45.30	51.47	53.09
15	MD-15	0.518	42.54	49.05	51.19
16	MD-16	0.444	52.03	56.26	59.19
17	MD-17	0.467	48.74	53.42	55.03
18	MD-18	0.489	46.59	53.21	53.67
19	MD-19	0.421	54.49	58.65	63.07
20	MD-20	0.442	53.06	56.67	62.57
21	MD-21	0.463	49.18	54.04	57.10
22	MD-22	0.518	40.02	46.92	48.48
23	MD-23	0.489	45.25	50.43	53.09
24	MD-24	0.511	42.68	48.54	49.63
25	MD-25	0.463	48.67	53.48	56.50
26	MD-26	0.484	45.52	50.97	53.63
27	MD-27	0.514	39.52	43.31	46.13

28	MD-28	0.543	31.66	37.18	43.92
29	MD-29	0.475	42.73	48.23	52.23
30	MD-30	0.507	40.69	44.46	46.42
31	MD-31	0.450	47.99	52.95	55.51
32	MD-32	0.475	44.89	51.20	53.85
33	MD-33	0.423	51.25	57.55	59.50
34	MD-34	0.447	49.05	54.14	57.35
35	MD-35	0.422	53.69	57.77	59.89
36	MD-36	0.543	36.64	43.46	46.55
37	MD-37	0.507	41.57	46.81	50.04
38	MD-38	0.475	46.22	52.58	53.07
39	MD-39	0.447	50.35	56.02	58.32
40	MD-40	0.422	54.11	58.52	62.28
41	MD-41	0.533	37.30	43.51	46.63
42	MD-42	0.500	44.04	50.53	52.55
43	MD-43	0.525	39.61	46.09	48.17
44	MD-44	0.471	47.37	51.31	54.77
45	MD-45	0.494	44.69	50.69	52.75
46	MD-46	0.444	50.93	55.71	59.05
47	MD-47	0.467	48.08	52.63	55.61
48	MD-48	0.421	54.14	58.21	61.11
49	MD-49	0.442	51.31	56.37	59.51

Table 3.7: Details of proportions for concrete mixes with 15% FA

S. No.	Mix designation	*w/cm*	Mix proportions (*cm* : FA : *fa* :*ca*)	Cement content (*c*) Kg/m^3	Fly ash content (FA) Kg/m^3
1	MDF-1	0.450	1:0.15:1.10:2.70	400	60.00
2	MDF-2	0.423	1:0.15:0.98:2.46	425	63.75
3	MDF-3	0.447	1:0.15:1.09:2.68	425	63.75
4	MDF-4	0.471	1:0.15:1.28:2.58	425	63.75
5	MDF-5	0.422	1:0.15:0.98:2.45	450	67.50
6	MDF-6	0.444	1:0.15:1.14:2.35	450	67.50
7	MDF-7	0.467	1:0.15:1.25:2.54	450	67.50
8	MDF-8	0.421	1:0.15:1.05:2.19	475	71.25
9	MDF-9	0.442	1:0.15:1.19:2.46	475	71.25
10	MDF-10	0.447	1:0.15:1.09:2.68	425	63.75
11	MDF-11	0.471	1:0.15:1.28:2.58	425	63.75
12	MDF-12	0.422	1:0.15:0.98:2.45	450	67.50
13	MDF-13	0.444	1:0.15:1.14:2.35	450	67.50
14	MDF-14	0.467	1:0.15:1.25:2.54	450	67.50
15	MDF-15	0.489	1:0.15:1.37:2.73	450	67.50

16	MDF-16	0.421	1:0.15:1.05:2.19	475	71.25
17	MDF-17	0.442	1:0.15:1.19:2.46	475	71.25
18	MDF-18	0.463	1:0.15:1.23:2.51	475	71.25
19	MDF-19	0.471	1:0.15:1.19:2.03	425	63.75
20	MDF-20	0.444	1:0.15:1.07:1.86	450	67.50
21	MDF-21	0.467	1:0.15:1.17:2.00	450	67.50
22	MDF-22	0.489	1:0.15:1.29:1.86	450	67.50
23	MDF-23	0.511	1:0.15:1.39:1.98	450	67.50
24	MDF-24	0.421	1:0.15:0.95:1.68	475	71.25
25	MDF-25	0.442	1:0.15:1.06:1.84	475	71.25
26	MDF-26	0.463	1:0.15:1.17:1.72	475	71.25
27	MDF-27	0.484	1:0.15:1.26:1.83	475	71.25

Table 3.8: Details of compressive strength of concrete mixes with 15% FA

S. No.	Mix designation	*w/cm*	f_{c28} (MPa)	f_{c56} (MPa)	f_{c91} (MPa)
1	MDF-1	0.450	39.04	47.65	52.20
2	MDF-2	0.423	45.09	50.21	55.75
3	MDF-3	0.447	41.14	48.67	52.69
4	MDF-4	0.471	38.35	43.27	50.42
5	MDF-5	0.422	46.13	51.01	56.51
6	MDF-6	0.444	42.50	49.17	53.11
7	MDF-7	0.467	39.58	44.02	51.07
8	MDF-8	0.421	47.34	52.30	57.70
9	MDF-9	0.442	43.55	49.79	53.79
10	MDF-10	0.447	42.01	49.68	53.39
11	MDF-11	0.471	38.85	44.96	50.50
12	MDF-12	0.422	47.25	51.95	57.17
13	MDF-13	0.444	43.09	50.30	53.67
14	MDF-14	0.467	40.26	45.30	51.62
15	MDF-15	0.489	37.15	44.52	48.08
16	MDF-16	0.421	48.41	53.58	58.19
17	MDF-17	0.442	44.02	51.81	54.12
18	MDF-18	0.463	40.73	45.95	52.10
19	MDF-19	0.471	38.90	43.20	50.50
20	MDF-20	0.444	43.22	49.93	53.62
21	MDF-21	0.467	39.85	44.61	51.42
22	MDF-22	0.489	36.87	41.25	47.30
23	MDF-23	0.511	35.23	40.05	46.11
24	MDF-24	0.421	47.94	53.05	57.82
25	MDF-25	0.442	43.87	50.48	54.38
26	MDF-26	0.463	40.34	45.61	52.39
27	MDF-27	0.484	37.65	42.28	48.55

3.3.1 General

If the observed variables are x_1, x_2, … ,x_n, the common factors are f_1, f_2, … , f_m and the unique factors are u_1, u_2, … , u_n, the variables may be expressed as linear functions of the factors.

$$x_1 = a_{11}f_1 + a_{12}f_2 + a_{13}f_3 + \cdots + a_{1m}f_m + a_1u_1$$

$$x_2 = a_{21}f_1 + a_{22}f_2 + a_{23}f_3 + \cdots + a_{2m}f_m + a_2u_2$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$x_n = a_{n1}f_1 + a_{n2}f_2 + a_{n3}f_3 + \cdots + a_{nm}f_m + a_nu_n \qquad (3.1)$$

Each one of the above equation is a regression equation; factor analysis seeks to find the coefficients a_{11}, a_{12}, … , a_{nm} which reproduce the observed variables from the factors. The coefficients are weights in the same manner as regression coefficients. In factor analysis, the coefficients are known as loadings and when the factors are uncorrelated, they reveal the correlation between each variable and a given factor. The above described set of equations is also known as factor analysis model.

3.3.2 Outputs from the dataset analysis

a) **Kaiser-Meyer-Olkin and Barlett's test:** Table 3.9 presents the results of Kaiser-Meyer-Olkin measure of sampling adequacy (KMO) and Bartlett's test of sphericity. The results of these tests are generally considered together as these tests provide a minimum standard which should be passed before factor analysis is to be carried out. KMO test gives a measure of sample size adequacy, *i.e.,* whether the sample size is large enough to apply factor analysis. The value of KMO test statistics, as derived, should lie between of 0 and 1 and it is considered essential for any dataset analysis that KMO statistic greater than 0.60. Bartlett's test of sphericity is statistical test for identifying the presence of correlation among the variables and a statistical significant Bartlett's test should the value of $p < 0.05$. The Bartlett's test also evaluates whether the correlation matrix is an identity matrix or not. For pre-investigation of dataset, the dataset has been taken in two different sets, F1 and F2. These sets are defined as:

$$F1 = \{w, cm, fa, ca, f_{c28}, f_{c56}, f_{c91}\} \text{ and}$$

$$F2 = \{w/\text{cm}, fa/cm, ca/cm, f_{c28}, f_{c56}, f_{c91}\}.$$

Table 3.9(a) presents results of KMO and Bartlett's test of sphericity for F1 and Table 3.9(b) represents results of KMO and Bartlett's test of sphericity for F2.

Table 3.9(a): Kaiser-Meyer-Olkin and Bartlett's Test of Sphericity

Kaiser-Meyer-Olkin Measure of Sampling Adequacy		0.646
Bartlett's Test of Sphericity	Approx. Chi-Square	704.917
	Df	21
	Sig.	0

It is observed from Table 3.9(a) and Table 3.9(b) that the values of KMO are 0.646 and 0.761 for F1 and F2, respectively indicating that both data formats can be considered for factor analysis. Higher KMO values indicate a better expectation of excellent results. As such, for further factor analysis, we will consider only F2 that has KMO a value greater than the KMO value of F1.

Table 3.9(b): Kaiser-Meyer-Olkin and Bartlett's Test of Sphericity

Kaiser-Meyer-Olkin Measure of Sampling Adequacy		0.761
Bartlett's Test of Sphericity	Approx. Chi-Square	540.632
	Df	15
	Sig.	0

b) Correlation matrix: A factor loading is interpreted as the Pearson correlation coefficient of an original variable with a factor. As correlations, the loadings have same domain of values from -1 (perfect negative association) to +1 (perfect positive association). Variables classically will have loadings on all factors but generally have high loadings on just one factor. Factor loadings that are below 0.30 may be ignored and the loadings are appreciated if the value is above 0.70.

Table 3.10(a) exhibits factor correlation matrix, which presents correlation between the factors extracted. It can be observed from the table that correlation between factor 1 and factor 2 is 0.350, which is near to the minimum value of factor loadings. It means that *fa*/*cm* has very small impact on prediction of compressive strength of concrete. The correlation between the Factor 1 and Factor 3 is 0.090, which is even less than 0.3 and has not any direct accountability for prediction of compressive strength of concrete. The correlation between Factor 1 and Factor 4 is -0.714, between Factor 1 and Factor 5 is -0.705 and between Factor 1 and Factor 6 is -0.850. Factor 4, Factor 5 and Factor 6

represents compressive strength of concrete at the curing ages of 28, 56 and 91 days, respectively. Table 3.10(a) also reveals that there is negative relationship of Factor 1 with Factor 4, Factor 5 and Factor 6. It means that if there is an increase in Factor 1, respective compressive strength (Factor 4, Factor 5 and Factor 6) of concrete will decrease accordingly.

Table 3.10(b) indicates anti image correlation matrix. This is used to measure the sample adequacy of individual variables. The diagonal elements of the matrix should be greater than 0.5 and if any of the diagonal element is less than 0.5, then it need to be removed from the analysis. In Table 3.10(b), all the elements are greater than 0.712 indicating that data is adequate.

Table 3.10(a): Correlation matrix

Factors		**1**	**2**	**3**	**4**	**5**	**6**
	Correlation	***w/cm***	***fa/cm***	***ca/cm***	f_{c28}	f_{c56}	f_{c91}
1	***w/cm***	1.000	0.358	0.090	-0.714	-0.705	-0.850
2	***fa/cm***	0.358	1.000	0.164	-0.497	-0.454	-0.437
3	***ca/cm***	0.090	0.164	1.000	-0.179	-0.153	-0.143
4	f_{c28}	-0.714	-0.497	-0.179	1.000	0.977	0.948
5	f_{c56}	-0.705	-0.454	-0.153	0.977	1.000	0.940
6	f_{c91}	-0.850	-0.437	-0.143	0.948	0.940	1.000

Table 3.10(b): Anti-image correlation matrix

	w/cm	***fa/cm***	***ca/cm***	f_{c28}	f_{c56}	f_{c91}
w/cm	0.712	-0.142	0.009	-0.287	-0.162	0.792
fa/cm	-0.142	0.836	-0.067	0.304	-0.112	-0.166
ca/cm	0.009	-0.067	0.810	0.120	-0.078	-0.028
f_{c28}	-0.287	0.304	0.120	0.746	-0.699	-0.472
f_{c56}	-0.162	-0.112	-0.078	-0.699	0.813	-0.241
f_{c91}	0.792	-0.166	-0.028	-0.472	-0.241	0.744

c) Communalities: Communality is the total amount of variance an original variable shares with all other variables included in the analysis. As a property of a particular variable, a very little communality indicates that variable is common with other variables in dataset and it is not worth considering it in the analysis. In Table 3.11, the initial is the variance in a variable at the beginning and extraction is the amount of variation retained out of total variance. If during the process, there is any dimension reduction (certain loss of information), then extraction variance is always less than the initial variance. From Table 3.11, it can be noted that 100% of variance associated with

w/cm is explained by the underlying factors and the same is with other variables of dataset. It simply means that every variable has some property to share with other variables and as an individual variable, these variables are not accountable for the present task, but as a dataset or together, they are contributing 100% to achieve the desired results.

d) Total variance explained: Table 3.12(a) shows the factors extracted and eigen values associated with each factor. The eigen value of a factor represent the variance explained by that factor. As there were six factors, therefore the total variance was also six. The first factor with eigen value of 3.88 explains 64.70% of total variance.

Table 3.11: Communalities

	Initial	**Extraction**
w/cm	1.00	1.00
fa/cm	1.00	1.00
ca/cm	1.00	1.00
f_{c28}	1.00	1.00
f_{c56}	1.00	1.00
f_{c91}	1.00	1.00

The second factor with eigen value of 0.99 explains 16.52% variance, third factor with eigen value of 0.72 explains 11.92% variance. Factor 4 with eigen value 0.37 explains 6.05% of the total variance, whereas Factor 5 and Factor 6 has negligible function in the total explained variance. A total variance of 99.19% variance is explained by the first four factors and 64.70% of the total variance is only explained by factor 1 *i.e., w/cm* (as shown in the column of 'cumulative %'). In this analysis, we have instructed to retain all the factors having eigen value greater than 0, therefore, column named 'extraction sums of squared loadings' has six rows, *i.e.,* initial eigen values and extracted sums of squared loadings are representing same number of factors in Table 3.12(a).

Table 3.12(a): Total variance explained

Component	**Initial Eigen values**			**Extraction Sums of Squared Loadings**		
	Total	**% of Variance**	**Cumulative %**	**Total**	**% of Variance**	**Cumulative %**
1	3.88	64.70	64.70	3.88	64.70	64.70
2	0.99	16.52	81.23	0.99	16.52	81.23
3	0.72	11.92	93.15	0.72	11.92	93.15

4	0.36	6.05	99.19	0.36	6.05	99.19
5	0.03	0.47	99.67	0.03	0.47	99.67
6	0.02	0.33	100.00	0.02	0.33	100.00

Table 3.12(b) indicates rotated total variance explained. Varimax method is used here for the rotation. This method helps to maximize the variance of each factor.

Table 3.12(b): Rotated total variance explained

Component	Rotation Sums of Squared Loadings		
	Total	**% of Variance**	**Cumulative %**
1	2.7	46.15	46.15
2	1.10	18.36	64.50
3	1.0	17.81	82.32
4	1.01	16.84	99.16
5	0.03	0.48	99.65
6	0.02	0.36	100.00

As it is considered that total variance explained is the shared variance of all the extracted components. Therefore, after the extraction of components, the total amount of variance accounted is distributed again between the extracted components. After the rotation, first component explains 46.15% of the total variance, second component explains 18.36%, third component explains 17.81% and fourth component explains 16.84% of total variance. There is not any special impact of rotation on fifth and sixth components, *i.e.,* only 0.48% and 0.3% of the total variance is explained by these two components, respectively. Before rotation and after rotation, first component is the only component that explains a great part of the total variance, *i.e.,* 64.70% of total variance before rotation and 46.15% of the total variance after rotation.

e) Scree plot: A scree test or a scree plot is a graphical method for determining the number of components. A scree plot is the graphical representation of the magnitude of each and every eigen value on the vertical axis plotted against their component numbers on the horizontal axis. In order to determine the appropriate number of components to retain and interpret, it is appropriate to look for the bend in the line. A typical scree plot will show the first few eigen values to be relatively gigantic in magnitude, with the magnitude of successive eigen values dropping off rather drastically. At some point, the line will appear to level off. The recommendation usually is to retain all the components with eigen values in the sharp descent of the line before the first one where the leveling effect occurs. The scree plot for the analysis of dataset is shown in Fig. 3.1. This figure indicates the point of inflexion on the curve. This plot also indicates that Component 1

is explaining highest variance of dataset. The curve starts bending on Component 2, it means that other successive components are accounting for smaller and smaller amounts of the total variance. There is another minor drop after Component 5 before a stable plateau is reached, it means that Component 5 and Component 6 have almost no contribution or very less contribution in the total variance explained.

f) Component Analysis: Table 3.13 exhibits component loadings, which are the correlations between the different variables and the components as these are correlations, possible values range from -1 to +1.

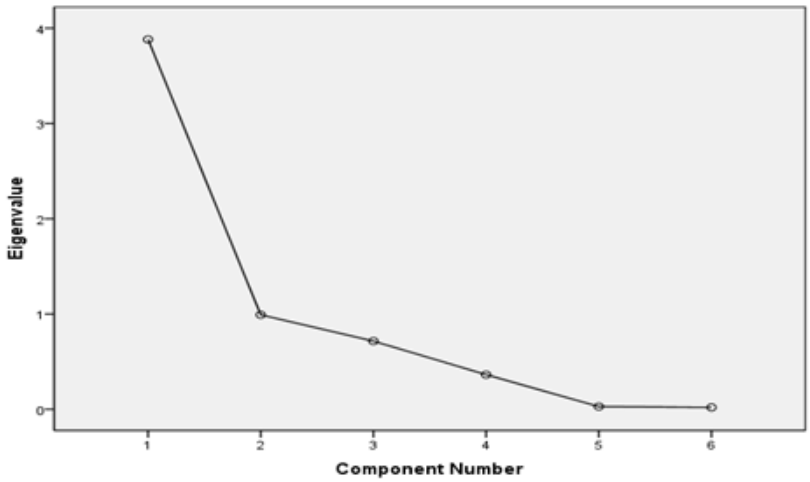

Fig. 3.1: Scree Plot

Higher the value of the loading, more the component contributes to the variable. This table shows that extracted Component 1, *i.e., w/cm* has greater impact on variables f_{c28} (compressive strength at age 28 days), *i.e.,* 96.50%, f_{c56} (compressive strength at age 56 days), *i.e.,* 95.40%, f_{c91} (compressive strength at age 91 days), *i.e.,* 97.50%. It is amply clear that *w/cm* contributes to a great extent to achieve desired compressive strength of concrete. Component 2 (*fa/cm*) contributes 94.20% to the variable *ca/cm* and Component 3 (*ca/cm*) contributes 77.30% for *fa/cm*. The loading having value less than 0.5 should be ignored.

Table 3.13: Component matrix

	Component					
	1	**2**	**3**	**4**	**5**	**6**
w/cm	-0.839	-0.162	-0.164	0.490	0.046	0.019
fa/cm	-0.581	0.242	0.773	0.074	-0.002	-0.007
ca/cm	-0.218	0.942	-0.251	0.037	0.001	-0.002
f_{c28}	0.965	0.037	0.049	0.228	0.015	-0.109

f_{c56}	0.954	0.071	0.083	0.253	-0.103	0.063
f_{c91}	0.975	0.109	0.134	0.002	0.125	0.058

3.3.3 Conclusions

In this chapter, factor analysis has been carried out to evaluate the impact of variables on the compressive strength of concrete for different curing ages. The findings of the study are as below:

- ✓ Dataset is adequate for the further analysis as KMO is equal to 76.10%.
- ✓ Data adequacy of each individual variable is greater than 71.20%.
- ✓ *w*/*cm* ratio comes out as a leading predictor variable as it is accounted for 64.70% of the total variance and *fa*/*cm* is the second predictor variable as it explains 16.52% of the total variance.
- ✓ *w*/*cm, fa*/*cm, ca*/*cm* and f_{c28} are the four factors that explain 99.19% of the total variance; f_{c56} and f_{c91} explain 0.81% of the total variance communally.
- ✓ Every factor is sharing its maximum variability with other variables of the dataset as communalities for all the components is 100%.
- ✓ *w*/*cm* has an impact of 96.50%, 95.40% and 97.5% on f_{c28}, f_{c56} and f_{c91}, respectively, as factor loadings are higher for these variables on Component 1.

The observations of the factor analysis reinforce the common knowledge among concrete engineers that the w/cm ratio is the predominant factor in deciding the compressive strength of concrete.

CHAPTER 4

REGRESSION MODELS FOR CONCRETE COMPRESSIVE STRENGTH PREDICTION

4.1 Introduction

Statistical models have the attraction that, once fitted, these can be used to perform predictions much more quickly than other modelling techniques and are simpler to implement in software (Zain, 2009). A number of research efforts have concentrated on using regression models to improve the accuracy of predictions. Popovics *et al.* (2008) augmented Abrams model, a widely accepted equation relating the water-cement ratio of concrete to its strength, with additional variables and used least square regression to determine the coefficients. Apart from its closed form expression, the other advantage of statistical modeling over other techniques is that, it is mathematically rigorous and can be used to define confidence interval for the predictions.

The most popular regression equation used by researchers for prediction of compressive strength is the linear regression equation:

$$f_c = a_o + a_1(\frac{w}{cm}) \tag{4.1}$$

where, f_c is compressive strength of concrete; $\frac{w}{cm}$ is water to cementitious material ratio; and a_0 and a_1 are regression coefficients. The origin of this equation is Abram's law, which states that in concrete materials, for a mixture of workable consistency, the strength of concrete is determined by the ratio of water to cementitious material (Popovics and Ujhelyi, 2008). According to this law, increasing the $\frac{w}{cm}$ ratio will lead to a decrease in concrete strength. The original formula for Abram's law is given in (4.2).

$$f_c = \frac{A}{B^{w/cm}} \tag{4.2}$$

where, f_c is compressive strength of concrete; $\frac{w}{cm}$ is the water to cementitious material ratio; and *A* and *B* are empirical constants. Lyse developed a formula similar to Abram's law using the ratio

of cementitious material and water contents instead of the ratio of water contents and cementitious material, as given in (4.3).

$$f_c = A + B(\frac{cm}{w}) \tag{4.3}$$

where, f_c is compressive strength of concrete; $\frac{cm}{w}$ is cementitious material to water content ratio; and *A* and *B* are empirical constants. Namyong *et al.* (2004) and Zain *et al.* (2009) further used this idea and proposed new methods for predicting the compressive strength of concrete.

The models proposed by Abram and Lyse did not account for the quantities of fine aggregates and coarse aggregates for prediction of compressive strength of concrete. So, for various concrete mixes, where $\frac{w}{cm}$ is constant, the strength will not change and this is not practically true. Thus, it is imperative to accommodate all the constituent materials into the predicting equation to have more reliable and accurate results for the prediction of concrete strength. For these reasons, Abram's law has been extended by various researchers, to include other variables in the form of multiple linear regression equation and are widely used to predict the compressive strength of various types of concrete. Such a general equation is given in (4.4).

$$f_c = b_0 + b_1\left(\frac{w}{cm}\right) + b_2\left(\frac{fa}{cm}\right) + b_3\left(\frac{ca}{cm}\right) \tag{4.4}$$

where, f_c is compressive strength of concrete; $\frac{W}{cm}$ is ratio of water content and cementitious material; $\frac{fa}{cm}$ is fine aggregate and cementitious material ratio; $\frac{ca}{cm}$ is coarse aggregate and cementitious material ratio; and $b_0, b_1, b_2\ and\ b_3$ are the multiple regression coefficients. As per this equation, all variables are linearly related to the compressive strength, but this may not always be true because the variables involved in a concrete mix and affecting the compressive strength are interrelated and the additive action does not always hold true. This highlights the need to look at alternative mathematical models that can reliably predict compressive strength of concrete with high accuracy. We have considered a relationship between different variables as,

$$f_c = a_0 x_1^{a1} x_2^{a2} x_3^{a3} \ldots x_m^m \tag{4.5}$$

The value of *m* changes from one model to other. Khedar et al. (2003) and Zain et al. (2009) have also proposed similar models. We have extended the study by considering different sets of dependent variables. Different regression techniques such as linear regression, non-linear and ridge regression have been used to study the variations in workability and fine aggregate on the compressive strength. The regression models have also been used to study the compressive strength of concrete with three different zones of coarse aggregates with and without FA. These models are presented in the sections of this chapter. The dataset used for the regression analysis for the prediction of compressive strength of concrete is given in Tables 3.5 - 3.8. In Section 4.2, we have presented the basic linear regression model for the sake of completeness.

4.2 Linear regression models for concrete compressive strength

Using the data given in Table 3.5 and Table 3.6, following linear models have been developed to predict the compressive strength of concrete.

$$f_{c28} = a_0 + a_1 cm + a_2 w + a_3 fa + a_4 ca \tag{4.6}$$

$$f_{c56} = a_0 + a_1 cm + a_2 w + a_3 fa + a_4 ca + a_5 f_{c28} \tag{4.7}$$

$$f_{c91} = a_0 + a_1 cm + a_2 w + a_3 fa + a_4 ca + a_5 f_{c28} + a_6 f_{c56} \tag{4.8}$$

$$f_{c91,28} = a_0 + a_1 cm + a_2 w + a_3 fa + a_4 ca + a_5 f_{c28} \tag{4.9}$$

$$f_{c91,56} = a_0 + a_1 cm + a_2 w + a_3 fa + a_4 ca + a_5 f_{c56} \tag{4.10}$$

where, f_{c28} is compressive strength of concrete after 28 days of curing age, f_{c56} is compressive strength of concrete after 56 days of curing age, and f_{c91} is compressive strength of concrete after 91 days of curing age. $f_{c91,28}$ is compressive strength of concrete after 91 days of curing and $\frac{f_{c28}}{cm}$ is engaged as one of the independent variable. $f_{c91,56}$ is compressive strength of concrete after 91 days of curing and $\frac{f_{c56}}{cm}$ is engaged as one of the independent variable. In (4.6) to (4.10), in predicting strength for higher ages, the strength of concrete at lower ages has also been considered in the model developed.

Table 4.1 contains the values of parameters for these models. The coefficient of determination (R^2) and root mean square error of these models is also given in Table 4.1.

It can be observed from this table that the value of R^2 is higher than 95.00% for all curing ages and the highest value of R^2 is achieved for f_{c56}. The lowest value of R^2, in this table, is 95.40% (which is also a considerably good fit) for f_{c28}. One can also infer that linear regression equations are well

suited for all the curing ages when cement content is not replaced with FA. Table 4.2 shows the regression coefficients for predicting compressive strength of concrete with replacement of cement content with 15% FA. It has been observed that the value of R^2 is higher than 97.00% for prediction of compressive strength of concrete at all curing ages except for f_{c56}. In case of 56 days compressive strength of concrete, the value of R^2 is 92.90%. The highest value of R^2 (= 98.50%) achieved in this analysis is for f_{c91}.

Table 4.1: Linear regression models for the prediction of compressive strength of concrete without FA

Regression variables, R^2 and RMSE	Regression coefficients for dependent variables				
	f_{c28}	f_{c56}	f_{c91}	$f_{c91,28}$	$f_{c91,56}$
Constant	28.6000	-0.6900	22.4000	22.1000	23.5000
cm	0.1700	-0.0390	0.07340	0.0559	0.08280
w	-0.3100	0.09720	-0.1640	-0.1210	-0.1830
fa	0.0085	-0.00547	0.0099	0.0074	0.0106
ca	0.0005	0.00074	-0.0002	0.0001	-0.0002
f_{c28}	-	1.1100	0.1210	0.6180	-
f_{c56}	-	-	0.4480	-	0.5240
R^2	0.9540	0.9710	0.9690	0.9630	0.9680
RMSE	1.2200	0.9000	0.9405	1.0132	0.9331

Table 4.2: Linear regression models for the prediction of compressive strength of concrete with 15% FA

Regression variables, R^2 and RMSE	Regression coefficients for dependent variables				
	f_{c28}	f_{c56}	f_{c91}	$f_{c91,28}$	$f_{c91,56}$
Constant	25.0000	28.2000	36.8000	34.4000	45.6000
cm	0.2040	0.1380	0.0948	0.0835	0.1710
w	-0.2910	-0.2450	-0.1570	-0.1370	-0.2630
fa	-0.0035	0.0031	-0.0045	-0.0048	-0.0061
ca	-0.0001	0.0022	0.0007	0.0005	0.0005
f_{c28}	-	0.3060	0.4160	0.3910	-
f_{c56}	-	-	-0.0818	-	-0.0370
R^2	0.9710	0.9290	0.9850	0.9840	0.9790
RMSE	0.6763	1.1472	0.4442	0.4435	0.5178

4.3 Regression models for the prediction of compressive strength of concrete for varying workability /effect of workability on compressive strength

In the present study, a regression model is investigated for its performance as a model for predicting compressive strength of concrete. Moreover, the effects of changes in the coefficients of

regression model on the performance curve have also been examined. For this purpose, multiple regression analysis has been carried out for predicting compressive strength of concrete using four variables, namely, water-cementitious material ratio, fine aggregate-cementitious ratio, coarse aggregate-cementitious ratio and cementitious content. Regression models are developed for concrete with medium and high workability at different curing ages (28, 56 and 91days).

4.3.1 Dataset used

Tables 4.3 - 4.6 contain the data used in this experiment. The details of the mixes using different proportions of *w*, *cm*, *fa, ca*1 and *ca*2 are shown in Table 4.3 and Table 4.4. Table 4.5 and Table 4.6 contain the compressive strength for all curing ages, for medium and high workability concrete mixes, respectively.

Table 4.3: Details of proportions for concrete mixes for medium workability

S. No.	Mix designation	*w/cm* ratio	Mix proportions (*cm* : *fa* : *ca*1 : *ca*2)	Cement content (*c*), Kg/m^3	Workability and Vee-Bee time in seconds
1.	MDM-1	0.514	1:1.392:2.181:1.074	350	MED (7.2)
2.	MDM-2	0.543	1:1.497:2.294:1.130	350	MED (5.2)
3.	MDM-3	0.480	1:1.245:2.001:0.986	375	MED (7.5)
4.	MDM-4	0.507	1:1.354:2.134:1.051	375	MED (6.0)
5.	MDM-5	0.450	1:1.100:1.811:0.892	400	MED (7.8)
6.	MDM-6	0.475	1:1.210:1.953:0.962	400	MED (6.6)
7.	MDM-7	0.423	1:0.981:1.650:0.813	425	MED (8.0)
8.	MDM-8	0.447	1:1.087:1.794:0.883	425	MED (6.9)
9.	MDM-9	0.422	1:0.977:1.644:0.810	450	MED (7.3)
10.	MDM-10	0.543	1:1.497:1.712:1.712	350	MED (5.7)
11.	MDM-11	0.507	1:1.354:1.593:1.593	375	MED (6.4)
12.	MDM-12	0.475	1:1.210:1.458:1.458	400	MED (6.9)
13.	MDM-13	0.447	1:1.087:1.339:1.339	425	MED (7.2)
14.	MDM-14	0.422	1:0.977:1.226:1.226	450	MED (7.5)

Table 4.4: Details of proportions for concrete mixes for high workability

S. No.	Mix designation	*w/cm* ratio	Mix proportions (*cm* : *fa* : *ca*1 : *ca*2)	Cement content (*c*), Kg/m^3	Workability and Vee-Bee time in seconds
1.	MDH-1	0.533	1:1.581:2.042:1.006	375	HIGH (4.5)

2.	MDH-2	0.500	1:1.430:1.892:0.932	400	HIGH (5.0)
3.	MDH-3	0.525	1:1.543:2.005:0.988	400	HIGH (4.2)
4.	MDH-4	0.471	1:1.275:1.725:0.850	425	HIGH (5.2)
5.	MDH-5	0.494	1:1.397:1.858:0.915	425	HIGH (4.5)
6.	MDH-6	0.444	1:1.140:1.576:0.776	450	HIGH (5.6)
7.	MDH-7	0.467	1:1.254:1.702:0.839	450	HIGH (4.9)
8.	MDH-8	0.442	1:1.192:1.650:0.813	475	HIGH (5.1)
9.	MDH-9	0.533	1:1.581:1.524:1.524	375	HIGH (4.8)
10.	MDH-10	0.500	1:1.430:1.412:1.412	400	HIGH (5.0)
11.	MDH-11	0.525	1:1.543:1.497:1.497	400	HIGH (4.2)
12.	MDH-12	0.471	1:1.275:1.288:1.288	425	HIGH (5.6)
13.	MDH-13	0.518	1:1.511:1.473:1.473	425	HIGH (4.0)
14.	MDH-14	0.467	1:1.254:1.271:1.271	450	HIGH (5.2)
15.	MDH-15	0.442	1:1.192:1.232:1.232	475	HIGH (5.5)
16.	MDH-16	0.463	1:1.230:1.255:1.255	475	HIGH (4.8)

Table 4.5: Compressive strength data for concrete mixes with medium workability

Mix designation	Mean and SD of f_{c28} (in MPa)		Mean and SD of f_{c56} (in MPa)		Mean and SD of f_{c91} (in MPa)	
	Mean	SD	Mean	SD	Mean	SD
MDM-1	39.52	0.97	43.31	1.51	46.13	0.99
MDM-2	31.66	1.4	37.18	1.77	43.92	1.92
MDM-3	42.73	1.4	48.23	1.07	52.23	1.53
MDM-4	40.69	1.1	44.46	1.83	46.42	2.03
MDM-5	47.99	1.7	52.95	1.1	55.51	1.64
MDM-6	44.89	1.24	51.2	1.36	53.85	1.1
MDM-7	51.25	1.64	57.55	1.57	59.5	1.86
MDM-8	49.05	1.38	54.14	1.03	57.35	1.51
MDM-9	53.69	1.37	57.77	1.35	59.89	1.55
MDM-10	36.64	0.99	43.46	2.03	46.55	1.81
MDM-11	41.57	1.37	46.81	1.63	50.04	1.8
MDM-12	46.22	1.03	52.58	1.61	53.07	1.83
MDM-13	50.35	1.36	56.02	1.57	58.32	1.62
MDM-14	54.11	1.42	58.52	1.33	62.28	1.53

Table 4.6: Compressive strength data for concrete mixes with high workability

Mix designation	Mean and SD of f_{c28} (in MPa)		Mean and SD of f_{c56} (in MPa)		Mean and SD of f_{c91} (in MPa)	
	Mean	SD	Mean		Mean	SD
MDH-1	36.84	1.97	40.92	2.21	44.52	1.82
MDH-2	43.13	1.75	50.22	1.79	51.97	1.95
MDH-3	38.58	2.04	45.51	2.2	47.49	1.76
MDH-4	47.16	1.18	51.25	1.2	54.27	1.29
MDH-5	45.05	1.8	50.72	1.35	52.85	1.42

MDH-6	49.63	1.5	54.48	1.45	58.04	1.57
MDH-7	47.42	1.55	51.34	1.81	55.3	1.02
MDH-8	50.05	1.34	55.72	1.43	58.31	1.32
MDH-9	37.81	1.19	43.5	1.91	47.55	1.2
MDH-10	44.11	1.62	50.98	1.62	52.56	1.51
MDH-11	40.9	1.36	46.56	1.83	51.07	1.19
MDH-12	47.51	1.31	52.92	1.36	54.47	1.23
MDH-13	42.54	1.93	49.05	1.83	51.19	1.21
MDH-14	48.74	1.37	53.42	1.63	55.03	1.36
MDH-15	53.06	1.32	56.67	1.53	62.57	1.43
MDH-16	49.18	1.17	54.04	1.37	57.1	1.14

4.3.2 Proposed prediction models

Namyong *et al.* (2004) developed regression equations, wherein, the ratio of *cm* to total aggregates (*fa, ca*1 and *ca*2) has also been used. The equations developed by them and named as Model-1 in the present study, are given in (4.11) - (4.15).

$$f_{c28} = \exp\left[B_0 + B_1\left(\frac{w}{cm}\right) + B_2(cm) + B_3\left(\frac{cm}{fa+ca1+ca2}\right)\right] \quad (4.11)$$

$$f_{c56} = exp\left[B_0 + B_1\left(\frac{w}{cm}\right) + B_2(cm) + B_3\left(\frac{cm}{(fa+ca1+ca2)}\right) + B_4\left(\frac{cm}{f_{c28}}\right)\right] \quad (4.12)$$

$$f_{c91} = exp\left[B_0 + B_1\left(\frac{w}{cm}\right) + B_2(cm) + B_3\left(\frac{cm}{(fa+ca1+ca2)}\right) + B_4\left(\frac{cm}{f_{c28}}\right) + B_5\left(\frac{cm}{f_{c56}}\right)\right] \quad (4.13)$$

$$f_{c91,28} = exp\left[B_0 + B_1\left(\frac{w}{cm}\right) + B_2(cm) + B_3\left(\frac{cm}{(fa+ca1+ca2)}\right) + B_4\left(\frac{cm}{f_{c28}}\right)\right] \quad (4.14)$$

$$f_{c91,56} = exp\left[B_0 + B_1\left(\frac{w}{cm}\right) + B_2(cm) + B_3\left(\frac{cm}{(fa+ca1+ca2)}\right) + B_5\left(\frac{cm}{f_{c56}}\right)\right] \quad (4.15)$$

On the basis of Model-1, we have developed two new models Model-2 and Model-3. In Model-2, ratios of *w*, *fa, ca*1, and *ca*2 have been taken with *cm.* In Model-3, the ratio of *cm* is considered separately with both types of course aggregates. We have thus proposed following equations for Model-2.

$$f_{c28} = A_0\left(\frac{w}{cm}\right)^{A_1}\left(\frac{fa}{cm}\right)^{A_2}\left(\frac{ca1}{cm}\right)^{A_3}\left(\frac{ca2}{cm}\right)^{A_4} \quad (4.16)$$

$$f_{c56} = A_0\left(\frac{w}{cm}\right)^{A_1}\left(\frac{fa}{cm}\right)^{A_2}\left(\frac{ca1}{cm}\right)^{A_3}\left(\frac{ca2}{cm}\right)^{A_4}\left(\frac{f_{c28}}{cm}\right)^{A_5} \quad (4.17)$$

$$f_{c91} = A_0\left(\frac{w}{cm}\right)^{A_1}\left(\frac{fa}{cm}\right)^{A_2}\left(\frac{ca1}{cm}\right)^{A_3}\left(\frac{ca2}{cm}\right)^{A_4}\left(\frac{f_{c28}}{cm}\right)^{A_5}\left(\frac{f_{c56}}{cm}\right)^{A_6} \quad (4.18)$$

$$f_{c91,28} = A_0\left(\frac{w}{cm}\right)^{A_1}\left(\frac{fa}{cm}\right)^{A_2}\left(\frac{ca1}{cm}\right)^{A_3}\left(\frac{ca2}{cm}\right)^{A_4}\left(\frac{f_{c28}}{cm}\right)^{A_5} \quad (4.19)$$

$$f_{c91,56} = A_0\left(\frac{w}{cm}\right)^{A_1}\left(\frac{fa}{cm}\right)^{A_2}\left(\frac{ca1}{cm}\right)^{A_3}\left(\frac{ca2}{cm}\right)^{A_4}\left(\frac{f_{c56}}{cm}\right)^{A_5} \quad (4.20)$$

The regression equations for Model-3 are given below.

$$f_{c28} = exp\left[B_0 + B_1\left(\frac{w}{cm}\right) + B_2(cm) + B_3\left(\frac{cm}{fa+ca1}\right) + B_4\left(\frac{cm}{fa+ca2}\right)\right] \quad (4.21)$$

$$f_{c56} = exp\left[B_0 + B_1\left(\frac{w}{cm}\right) + B_2(cm) + B_3\left(\frac{cm}{(fa+ca1)}\right) + B_4\left(\frac{cm}{fa+ca2}\right) + B_5\left(\frac{cm}{f_{c28}}\right)\right] \quad (4.22)$$

$$f_{c91} = \exp\left[B_0 + B_1\left(\frac{w}{cm}\right) + B_2(cm) + B_3\left(\frac{cm}{(fa+ca1)}\right) + B_4\left(\frac{cm}{fa+ca2}\right) + B_5\left(\frac{cm}{f_{c28}}\right) + B_6\left(\frac{cm}{f_{c56}}\right)\right] \quad (4.23)$$

$$f_{c91,28} = \exp\left[B_0 + B_1\left(\frac{w}{cm}\right) + B_2(cm) + B_3\left(\frac{cm}{(fa+ca1)}\right) + B_4\left(\frac{cm}{fa+ca2}\right) + B_5\left(\frac{cm}{f_{c28}}\right)\right] \quad (4.24)$$

$$f_{c91,56} = \exp\left[B_0 + B_1\left(\frac{w}{cm}\right) + B_2(cm) + B_3\left(\frac{cm}{(fa+ca1)}\right) + B_4\left(\frac{cm}{fa+ca2}\right) + B_5\left(\frac{cm}{f_{c56}}\right)\right] \quad (4.25)$$

where, f_{c28} is compressive strength of concrete after 28 days of curing, f_{c56} is compressive strength of concrete after 56 days of curing, and f_{c91} is compressive strength of concrete after 91 days of curing. $f_{c91,28}$ is compressive strength of concrete after 91 days of curing when $\frac{f_{c28}}{cm}$ is also considered as one of the independent variable. $f_{c91,56}$ is compressive strength of concrete after 91 days of curing when $\frac{f_{c56}}{cm}$ is also considered as one of the independent variables.

The three models, as given above, have been used for analysis of experimental data given in Tables 4.3 - 4.6. The regression coefficients so obtained along with other related statistical parameters have been tabulated. Table 4.7 contains the regression coefficients for Model-1 obtained on the medium workability and Table 4.8 contains these coefficients for high workability. The regression coefficients obtained for Model-2 for medium and high workability mixes are provided in Table 4.9 and Table 4.10, respectively. Also, the regression coefficients obtained for Model-3 for medium and high workability mixes are provided in Table 4.11 and Table 4.12, respectively.

Table 4.7: Regression coefficients of multiple regression models predicting the compressive strength of concrete with medium workability for Model–1

Concrete Mixes with Medium Workability					
Coefficient	f_{c28}	f_{c56}	f_{c91}	$f_{c91,28}$	$f_{c91,56}$
B_0	5.4926	3.7079	3.4846	3.5614	3.4906
B_1	-3.6013	0.0003	0.7057	-0.4983	-0.1543
B_2	0.0004	0.0023	0.0024	0.0018	0.0020
B_3	-0.5411	0.1164	0.2124	0.4842	0.4839
B_4	-	-0.0830	-0.0703	0.0217	-0.0496
B_5	-	-	-0.0294	-	-

Table 4.8: Regression coefficients of multiple regression models predicting the compressive strength of concrete with high workability for Model–1

Concrete Mixes with High Workability					
Coefficient	f_{c28}	f_{c56}	f_{c91}	$f_{c91,28}$	$f_{c91,56}$
B_0	5.0848	3.4319	3.4846	3.3753	3.9898
B_1	-2.9811	1.0184	0.7057	0.9214	-0.4099
B_2	0.0006	0.0024	0.0023	0.0023	0.0019
B_3	-0.4116	0.1175	0.2123	0.2332	0.0671
B_4	-	-0.1164	-0.0703	-0.0976	-0.0754
B_5	-	-	-0.0294	-	-

4.3.3 Results and discussions

In order to establish the effectiveness of the proposed models we have calculated the coefficient of determination and root mean square error for these models. These are given in Tables 4.13 and 4.14 for medium and high workability mixes, respectively. Table 4.13 contains the results on medium workability for Model-1, Model-2 and Model-3.In the analysis of Model-1, it has been observed that the value of R^2 is higher than 94.21% for all curing ages, indicating the suitability of the assumed regression equations. On further analysis, it is observed that the value of R^2 achieved for prediction of f_{c28} is the lowest (94.21%), which is increased for f_{c56} (97.57%) and it is 96.54% for prediction of f_{c91}.

Table 4.9: Regression coefficients of multiple regression models predicting the compressive strength of concrete with medium workability for Model–2

Concrete Mixes with Medium Workability					
Coefficient	f_{c28}	f_{c56}	f_{c91}	$f_{c91,28}$	$f_{c91,56}$
A_0	0.0222	89.3373	9.4989	32.3254	105.1563
A_1	-9.0712	-0.7887	-2.1096	-0.6680	0.3007
A_2	4.0255	-0.2764	0.3807	-0.3803	-0.8739
A_3	0.1424	0.0111	-0.0107	-0.0374	-0.0390
A_4	0.2142	0.1105	0.0768	0.0719	0.0551
A_5	-	0.5218	-0.5556	-0.0340	0.1399
A_6	-	-	0.5524	-	-

Table 4.10: Regression coefficients of multiple regression models predicting the compressive strength of concrete with high workability for Model–2

Concrete Mixes with High Workability					
Coefficient	f_{c28}	f_{c56}	f_{c91}	$f_{c91,28}$	$f_{c91,56}$

A_0	17.3445	31.1750	11.5347	10.8133	11.9447
A_1	-1.4096	-1.5687	-2.2648	-2.3694	-2.2904
A_2	-0.0302	0.3297	0.7271	0.8466	0.7662
A_3	-0.1262	-0.0752	-0.1189	-0.1221	-0.1186
A_4	0.0099	0.0073	-0.0028	-0.0053	-0.0036
A_5	-	0.3201	-0.0982	0.1417	0.1584
A_6	-	-	-0.2312	-	-

Table 4.11: Regression coefficients of multiple regression models predicting the compressive strength of concrete with medium workability for Model–3

Concrete Mixes with Medium Workability					
Coefficient	f_{c28}	f_{c56}	f_{c91}	$f_{c91,28}$	$f_{c91,56}$
B_0	6.9549	4.5112	4.3499	4.5507	3.8336
B_1	-5.3138	-1.1579	-1.4540	-1.9211	-0.6728
B_2	-0.0004	0.0016	0.0013	0.0009	0.0016
B_3	-0.3221	0.0279	0.1660	0.2186	0.2598
B_4	-0.7736	-0.2334	-0.0662	-0.1884	0.0185
B_5	-	-0.0672	0.0551	-0.0013	-0.0372
B_6	-	-	-0.0916	-	-

Table 4.12: Regression coefficients of multiple regression models predicting the compressive strength of concrete with high workability for Model–3

Concrete Mixes with High Workability					
Coefficient	f_{c28}	f_{c56}	f_{c91}	$f_{c91,28}$	$f_{c91,56}$
B_0	4.8570	3.3476	3.4957	3.3683	3.9523
B_1	-2.6876	1.3105	0.6762	0.9507	-0.4263
B_2	0.0004	0.0028	0.0023	0.0024	0.0017
B_3	0.3539	-0.1528	0.0989	0.0700	0.2351
B_4	-0.0908	0.0407	0.0527	0.0603	0.0163
B_5	-	-0.1309	-0.0685	-0.0989	-0.0680
B_6	-	-	-0.0297	-	-

Table 4.13: Comparison of models for medium workability mixes

Strength	**Model-1**		**Model-2**		**Model-3**	
	R^2	**RMSE**	R^2	**RMSE**	R^2	**RMSE**
f_{c28}	0.9421	0.0356	0.9905	0.0236	0.9650	0.0277
f_{c56}	0.9757	0.0203	0.9746	0.0208	0.9795	0.0186
f_{c91}	0.9654	0.0155	0.9838	0.0138	0.9889	0.0114
$f_{c91,28}$	0.9641	0.0205	0.9766	0.0166	0.9743	0.0174
$f_{c91,56}$	0.9779	0.0161	0.9781	0.0161	0.9805	0.0151

Table 4.14: Comparison of models for high workability mixes

Strength	Model-1		Model-2		Model-3	
	R^2	RMSE	R^2	RMSE	R^2	RMSE
f_{c28}	0.9492	0.0238	0.9688	0.0227	0.9632	0.0202
f_{c56}	0.9505	0.0196	0.8892	0.0292	0.9532	0.0190
f_{c91}	0.9654	0.0155	0.9334	0.0216	0.9656	0.0155
$f_{c91,28}$	0.9622	0.0162	0.9310	0.0219	0.9625	0.0162
$f_{c91,56}$	0.9528	0.0182	0.9330	0.0216	0.9566	0.0174

It can also be seen from the analysis that for prediction of f_{c91}, the value of R^2 is the highest (97.79%) when f_{c28} and f_{c56} are also considered in the development of the regression model, in addition to other parameters laid down in the equations.

On analysing the results for Model-2, one can observe that the value of R^2 is higher than 97.46% for all curing ages, indicating the suitability of the derived regression equations. The highest value of R^2 (99.05%) is achieved for f_{c28}, which is decreased for f_{c56} (97.46%); the value of R^2 for prediction of f_{c91} is 98.37%. To summarize, it can be inferred that the regression equations as presumed under Model-2 are best suited for f_{c28} for medium workability mixes. It can be seen that value of R^2 is higher than 96.50% for all curing ages when we employ Model-3. The highest value of R^2 (98.89%) is achieved for f_{c91}, which is decreased to 97.95% for f_{c56} and 96.50% for f_{c28}. As such, it can be concluded that the regression equations as per Model–3 are best suited to predict f_{c91} for medium workability mixes.

Table 4.14 contains the results of high workability for Model-1, Model-2 and Model-3. In the analysis of Model-1, it has been observed that the value of R^2 is higher than 94.00% for all curing ages. Again, as per the trend observed for other models with medium workability mixes, the highest value of R^2 (96.54%) is achieved for f_{c91}, which declines to 95.05% for f_{c56}, and 94.92% for f_{c28}, although these values of R^2 are less than the values achieved for medium workability mixes. Similarly, it can also be seen that for f_{c91}, the value of R^2 is highest when f_{c28} and f_{c56} are also considered in the development of the regression model.

The figures presented in Table 4.14 also reveal that the value of R^2 for Model-2 is higher than 93.10% for all curing ages except for f_{c56}, indicating the suitability of the presumed regression equations for f_{c28} and f_{c91}. Again, as the trend observed for medium workability mixes, the

highest value of R^2 of 96.88% is achieved for f_{c28}, which decreases for f_{c56} (88.92%). The value of R^2, however, increases for the prediction of f_{c91} (93.34%). These values of R^2 are less than those achieved for medium workability mixes.

While analysing the results of Model-3, it has been observed that value of R^2 is higher than 95.32% for all curing ages again, indicating the suitability of the assumed regression equations. As the trend observed for medium workability mixes, the highest value of R^2 (96.32%) is achieved for f_{c28}, which is reduced for f_{c56} (95.32%), but further it is increased for f_{c91} (96.56%), however, these coefficients are less than the ones achieved for medium workability mixes.

On comparing the regression analysis results achieved using the Model-2 and Model-3 and model as suggested by Namyong *et al.* (2004) *i.e.,* Model-1 as per the details provided in Table 4.13 and Table 4.14, it can be said that in general both Model-2 and Model-3 as proposed in the presented study provide better values of R^2 for both medium and high workability mixes. For medium workability mixes, Model-2 is found to be the best suited for f_{c28}, whereas, for f_{c56} and f_{c91}, both Model-2 and Model-3 are equally suitable. It is also observed that f_{c56} and f_{c91} can be best predicted if in addition to other parameters, the strength at the preceding ages is also known.

For high workability mixes, Model-2 is again found to be the best suited for f_{c28}, whereas, for f_{c56} and f_{c91}, both Model-1 and Model-3 are equally suitable. On similar lines as for medium workability mixes, it is observed that f_{c56} and f_{c91} can be best predicted if in addition to other parameters, the strength at the preceding ages is also known. The variation in the values of R^2 is, however, less for high workability mixes.

4.3.4 Conclusions

The work presented above comprises of the development of regression models for predicting compressive strength of concrete at three stages of its curing. The regression models thus developed can reliably predict compressive strength of various mixes efficiently. In general, both Model-2 and Model-3 provide better values of R^2 for medium and high workability mixes. For better prediction of f_{c91} for both medium and high workability mixes, it is recommended to consider f_{c28} and f_{c56} in the regression equations. The regression equation as per Model-2 is the best suited scheme for f_{c28} for medium workability mixes, whereas, the regression equation as per Model-3 is best suited scheme for f_{c91} for medium workability mixes, wherein, f_{c28} and f_{c56} are also included in regression equation. The regression equation as per Model-3 is best suited method

for predicting compressive strength of high workability mixes at all ages as compared to the other regression models.

4.4 Regression models for the prediction of compressive strength of concrete/ effect of flyash on compressive strength

4.4.1 Dataset used

The details of proportions for concrete mixes without any replacement of cement with FA and aggregates belonging to Zone-A, Zone-B and Zone-C are provided in Table 4.15, Table 4.17 and Table 4.19, respectively and details of compressive strength of concrete mixes of Zone-A, Zone-B and Zone-C aggregates without FA at curing ages of 28, 56 and 91 days, are presented in Table 4.16, Table 4.18 and 4.20, respectively. The details of proportions for concrete mixes with 15% replacement of cement with FA and aggregates belonging to aggregates of Zone-A, Zone-B and Zone-C are provided in Table 4.21, Table 4.23 and Table 4.25, respectively and details of compressive strength of concrete mixes with 15% FA with aggregates belonging to Zone-A, Zone-B and Zone-C at curing ages of 28, 56 and 91 days, are presented in Table 4.22, Table 4.24 and 4.25, respectively.

4.4.2 Proposed prediction model

The regression equations for proposed Model are given below.

$$f_{c28} = A_0 \left(\frac{w}{cm}\right)^{A_1} \left(\frac{fa}{cm}\right)^{A_2} \left(\frac{ca}{cm}\right)^{A_3} \tag{4.26}$$

$$f_{c56} = A_0 \left(\frac{w}{cm}\right)^{A_1} \left(\frac{fa}{cm}\right)^{A_2} \left(\frac{ca}{cm}\right)^{A_3} \left(\frac{f_{c28}}{cm}\right)^{A_4} \tag{4.27}$$

$$f_{c91} = A_0 \left(\frac{w}{cm}\right)^{A_1} \left(\frac{fa}{cm}\right)^{A_2} \left(\frac{ca}{cm}\right)^{A_3} \left(\frac{f_{c28}}{cm}\right)^{A_4} \left(\frac{f_{c56}}{cm}\right)^{A_5} \tag{4.28}$$

$$f_{c91,28} = A_0 \left(\frac{w}{cm}\right)^{A_1} \left(\frac{fa}{cm}\right)^{A_2} \left(\frac{ca}{cm}\right)^{A_3} \left(\frac{f_{c28}}{cm}\right)^{A_4} \tag{4.29}$$

$$f_{c91,56} = A_0 \left(\frac{w}{cm}\right)^{A_1} \left(\frac{fa}{cm}\right)^{A_2} \left(\frac{ca}{cm}\right)^{A_3} \left(\frac{f_{c56}}{cm}\right)^{A_4} \tag{4.30}$$

where, $\frac{w}{cm}$ is water-cementitious material ratio, $\frac{fa}{cm}$ is fine aggregate-cementitious material ratio and $\frac{ca}{cm}$ is the coarse aggregate-cementitious material ratio, f_{c28} is compressive strength of concrete at curing age of 28 days, f_{c56} is compressive strength of concrete at curing age of 56 days and f_{c91} compressive strength of concrete at curing age of 91 days. $f_{c91,28}$ is compressive strength of

concrete after 91 days of curing when $\frac{f_{c28}}{cm}$ is also considered as one of the independent variable. $f_{c91,56}$ is compressive strength of concrete after 91 days of curing when $\frac{f_{c56}}{cm}$ is also considered as one of the independent variables.

Table 4.15: Details of proportions for Concrete mixes with Zone-A of aggregates without FA

S. No.	Mix designation	*w/cm* ratio	Mix proportions (*cm* : *fa* : *ca*)	Cement content(*c*) Kg/m^3
1	MD-1	0.514	1:1.39:3.26	350
2	MD-2	0.543	1:1.49:3.42	350
3	MD-3	0.48	1:1.24:2.98	375
4	MD-4	0.507	1:1.35:3.18	375
5	MD-5	0.533	1:1.58:3.05	375
6	MD-6	0.45	1:1.10:2.70	400
7	MD-7	0.475	1:1.21:2.91	400
8	MD-8	0.5	1:1.43:2.82	400
9	MD-9	0.525	1:1.54:2.99	400
10	MD-10	0.423	1:0.98:2.47	425
11	MD-11	0.447	1:1.08:2.68	425
12	MD-12	0.471	1:1.27:2.57	425
13	MD-13	0.494	1:1.39:2.77	425
14	MD-14	0.422	1:0.97:2.45	450
15	MD-15	0.444	1:1.14:2.35	450
16	MD-16	0.467	1:1.25:2.54	450
17	MD-17	0.421	1:1.045:2.1	475
18	MD-18	0.442	1:1.192:2.4	475

Table 4.16: Details of compressive strength of concrete mixes with Zone-A aggregates without FA

S. No.	Mix designation	*w/cm* ratio	f_{c28} (MPa)	f_{c56} (MPa)	f_{c91} (MPa)
1	MD-1	0.514	39.52	43.31	46.13
2	MD-2	0.543	31.66	37.18	43.92
3	MD-3	0.48	42.73	48.23	52.23
4	MD-4	0.507	40.69	44.46	46.42
5	MD-5	0.533	36.84	40.92	44.52
6	MD-6	0.45	47.99	52.95	55.51
7	MD-7	0.475	44.89	51.2	53.85
8	MD-8	0.5	43.13	50.22	51.97
9	MD-9	0.525	38.58	45.51	47.49
10	MD-10	0.423	51.25	57.55	59.5
11	MD-11	0.447	49.05	54.14	57.35

12	MD-12	0.471	47.16	51.25	54.27
13	MD-13	0.494	45.05	50.72	52.85
14	MD-14	0.422	53.69	57.77	59.89
15	MD-15	0.444	49.63	54.48	58.04
16	MD-16	0.467	47.42	51.34	55.3
17	MD-17	0.421	54.01	57.91	60.15
18	MD-18	0.442	50.05	55.72	58.31

Table 4.17: Details of proportions for Concrete mixes with Zone-B of aggregates without FA

S. No.	Mix designation	*w/cm* ratio	Mix proportions (*cm* : *fa* : *ca*)	Cement content (*c*) Kg/m^3
1	MD-19	0.543	1:1.49:3.42	350
2	MD-20	0.507	1:1.35:3.18	375
3	MD-21	0.533	1:1.58:3.04	375
4	MD-22	0.475	1:1.21:2.91	400
5	MD-23	0.5	1:1.43:2.82	400
6	MD-24	0.525	1:1.54:2.99	400
7	MD-25	0.447	1:1.08:2.67	425
8	MD-26	0.471	1:1.27:2.57	425
9	MD-27	0.494	1:1.39:2.77	425
10	MD-28	0.518	1:1.51:2.94	425
11	MD-29	0.422	1:0.97:2.45	450
12	MD-30	0.444	1:1.14:2.35	450
13	MD-31	0.467	1:1.25:2.54	450
14	MD-32	0.489	1:1.37:2.73	450
15	MD-33	0.421	1:1.04:2.19	475
16	MD-34	0.442	1:1.19:2.46	475
17	MD-35	0.463	1:1.23:2.51	475

Table 4.18: Details of compressive strength of concrete mixes with Zone-B aggregates without FA

S. No.	Mix designation	*w/cm* ratio	f_{c28} (MPa)	f_{c56} (MPa)	f_{c91} (MPa)
1	MD-19	0.543	36.64	43.46	46.55
2	MD-20	0.507	41.57	46.81	50.04
3	MD-21	0.533	37.81	43.5	47.55
4	MD-22	0.475	46.22	52.58	53.07
5	MD-23	0.5	44.11	50.98	52.56
6	MD-24	0.525	40.9	46.56	51.07
7	MD-25	0.447	50.35	56.02	58.32

8	MD-26	0.471	47.51	52.92	54.47
9	MD-27	0.494	45.3	51.47	53.09
10	MD-28	0.518	42.54	49.05	51.19
11	MD-29	0.422	54.11	58.52	62.28
12	MD-30	0.444	52.03	56.26	59.19
13	MD-31	0.467	48.74	53.42	55.03
14	MD-32	0.489	46.59	53.21	53.67
15	MD-33	0.421	54.49	58.65	63.07
16	MD-34	0.442	53.06	56.67	62.57
17	MD-35	0.463	49.18	54.04	57.1

Table 4.19: Details of proportions for Concrete mixes with Zone-C of aggregates without FA

S. No.	Mix designation	*w/cm* ratio	Mix proportions (*cm* : *fa* : *ca*)	Cement content(*c*) Kg/m^3
1	MD-36	0.533	1:1.47:2.41	375
2	MD-37	0.5	1:1.32:2.21	400
3	MD-38	0.525	1:1.43:2.36	400
4	MD-39	0.471	1:1.19:2.03	425
5	MD-40	0.494	1:1.29:2.17	425
6	MD-41	0.518	1:1.42:2.01	425
7	MD-42	0.444	1:1.06:1.85	450
8	MD-43	0.467	1:3:51:2.00	450
9	MD-44	0.489	1:1.28:1.86	450
10	MD-45	0.511	1:1.39:1.98	450
11	MD-46	0.421	1:0.95:1.68	475
12	MD-47	0.442	1:1.05:1.84	475
13	MD-48	0.463	1:1.16:1.71	475
14	MD-49	0.484	1:1.26:1.83	475

Table 4.20: Details of compressive strength of concrete mixes with Zone-C aggregates without FA

S. No.	Mix designation	*w/cm* ratio	f_{c28} (MPa)	f_{c56} (MPa)	f_{c91} (MPa)
1	MD-36	0.533	37.3	43.51	46.63
2	MD-37	0.5	44.04	50.53	52.55
3	MD-38	0.525	39.61	46.09	48.17
4	MD-39	0.471	47.37	51.31	54.77
5	MD-40	0.494	44.69	50.69	52.75
6	MD-41	0.518	40.02	46.92	48.48
7	MD-42	0.444	50.93	55.71	59.05

8	MD-43	0.467	48.08	52.63	55.61
9	MD-44	0.489	45.25	50.43	53.09
10	MD-45	0.511	42.68	48.54	49.63
11	MD-46	0.421	54.14	58.21	61.11
12	MD-47	0.442	51.31	56.37	59.51
13	MD-48	0.463	48.67	53.48	56.5
14	MD-49	0.484	45.52	50.97	53.63

Table 4.21: Details of proportions for Concrete mixes with Zone-A of aggregates and 15% FA

S. No.	Mix designation	*w/cm* ratio	Mix proportions (*cm* : *fa* : *ca*)	Cement content (*c*) Kg/m^3	Fly ash content (FA) Kg/m^3
1	MDF-1	0.450	1:1.10:2.70	400	60.00
2	MDF-2	0.423	1:0.98:2.46	425	63.75
3	MDF-3	0.447	1:1.08:2.68	425	63.75
4	MDF-4	0.471	1:1.27:2.57	425	63.75
5	MDF-5	0.422	1:0.97:2.45	450	67.50
6	MDF-6	0.444	1:1.14:2.35	450	67.50
7	MDF-7	0.467	1:1.25:2.54	450	67.50
8	MDF-8	0.421	1:1.04:2.19	475	71.25
9	MDF-9	0.442	1:1.19:2.46	475	71.25

Table 4.22: Details of compressive strength of concrete mixes with Zone-A aggregates with 15% FA

S. No.	Mix designation	*w/cm* ratio	f_{c28} (MPa)	f_{c56} (MPa)	f_{c91} (MPa)
1	MDF-1	0.450	39.04	47.65	52.2
2	MDF-2	0.423	45.09	50.21	55.75
3	MDF-3	0.447	41.14	48.67	52.69
4	MDF-4	0.471	38.35	43.27	50.42
5	MDF-5	0.422	46.13	51.01	56.51
6	MDF-6	0.444	42.5	49.17	53.11
7	MDF-7	0.467	39.58	44.02	51.07
8	MDF-8	0.421	47.34	52.3	57.7
9	MDF-9	0.442	43.55	49.79	53.79

Table 4.23: Details of proportions for Concrete mixes with Zone-B of aggregates and 15% FA

S. No.	Mix designation	*w/cm* ratio	Mix proportions (*cm* : *fa* : *ca*)	Cement content (*c*) Kg/m^3	Fly ash content (FA) Kg/m^3
1	MDF-10	0.447	1:1.08:2.67	425.00	63.75

2	MDF-11	0.471	1:1.27:2.57	425.00	63.75
3	MDF-12	0.422	1:0.97:2.45	450.00	67.5
4	MDF-13	0.444	1:1.14:2.35	450.00	67.5
5	MDF-14	0.467	1:1.25:2.54	450.00	67.5
6	MDF-15	0.489	1:1.37:2.73	450.00	67.5
7	MDF-16	0.421	1:1.04:2.19	475.00	71.25
8	MDF-17	0.442	1:1.19:2.46	475.00	71.25
9	MDF-18	0.463	1:1.23:2.51	475.00	71.25

Table 4.24: Details of compressive strength of concrete mixes with Zone-B aggregates and 15% FA

S. No.	Mix designation	*w/cm* ratio	f_{c28} (MPa)	f_{c56} (MPa)	f_{c91} (MPa)
1	MDF-10	0.447	42.01	49.68	53.39
2	MDF-11	0.471	38.85	44.96	50.50
3	MDF-12	0.422	47.25	51.95	57.17
4	MDF-13	0.444	43.09	50.30	53.67
5	MDF-14	0.467	40.26	45.30	51.62
6	MDF-15	0.489	37.15	44.52	48.08
7	MDF-16	0.421	48.41	53.58	58.19
8	MDF-17	0.442	44.02	51.81	54.12
9	MDF-18	0.463	40.73	45.95	52.1

Table 4.25: Details of proportions for Concrete mixes with Zone-C of aggregates and 15% FA

S. No.	Mix designation	*w/cm* ratio	Mix proportions (*cm* : *fa* : *ca*)	Cement content (*c*) Kg/m^3	Fly ash content (FA) Kg/m^3
1	MDF-19	0.471	1:1.19:2.03	425	63.75
2	MDF-20	0.444	1:1.06:1.85	450	67.50
3	MDF-21	0.467	1:1.17:2.00	450	67.50
4	MDF-22	0.489	1:1.28:1.86	450	67.50
5	MDF-23	0.511	1:1.39:1.98	450	67.50
6	MDF-24	0.421	1:0.95:1.68	475	71.25
7	MDF-25	0.442	1:1.05:1.84	475	71.25
8	MDF-26	0.463	1:1.16:1.71	475	71.25
9	MDF-27	0.484	1:1.26:1.83	475	71.25

Table 4.26: Details of compressive strength of concrete mixes with Zone-C aggregates and 15% FA

S. No.	Mix designation	*w/cm* ratio	f_{c28} (MPa)	f_{c56} (MPa)	f_{c91} (MPa)
1	MDF-19	0.471	38.90	43.20	50.50
2	MDF-20	0.444	43.22	49.93	53.62
3	MDF-21	0.467	39.85	44.61	51.42

4	MDF-22	0.489	36.87	41.25	47.30
5	MDF-23	0.511	35.23	40.05	46.11
6	MDF-24	0.421	47.94	53.05	57.82
7	MDF-25	0.442	43.87	50.48	54.38
8	MDF-26	0.463	40.34	45.61	52.39
9	MDF-27	0.484	37.65	42.28	48.55

Tables 4.27 - 4.32 contain the regression coefficients and R^2 for proposed model for Zone-A, Zone-B and Zone-C aggregates, cement without replacing FA and cement with FA replacement.

Table 4.27: Regression coefficients of multiple regression model predicting the compressive strength of concrete with Zone-A aggregates without FA

Concrete Mixes with Zone-A Aggregates					
Coefficient	f_{c28}	f_{c56}	f_{c91}	$f_{c91,28}$	$f_{c91,56}$
A_0	3.1688	15.0896	26.3395	14.1329	30.8196
A_1	-3.1459	-2.6449	-1.5201	-1.8367	-1.5043
A_2	0.7646	0.8674	0.2932	0.3850	0.3429
A_3	0.1286	0.0058	-0.1769	-0.0292	-0.1599
A_4	-	0.4406	-0.4689	0.0481	0.2355
A_5	-	-	0.6463	-	-
R^2	0.8866	0.8995	0.9430	0.9090	0.9233

Table 4.28: Regression coefficients of multiple regression models predicting the compressive strength of Concrete with Zone-A of aggregates with 15% FA

Concrete Mixes with Zone-A Aggregates					
Coefficient	f_{c28}	f_{c56}	f_{c91}	$f_{c91,28}$	$f_{c91,56}$
A_0	8.4570	1.8925	17.8367	18.4700	18.7979
A_1	-2.1654	-2.4073	-1.0491	-1.0127	-1.0562
A_2	0.2990	0.0837	-0.0544	-0.0716	-0.0355
A_3	-0.2019	0.2156	-0.0703	-0.1155	-0.0669
A_4	-	-0.4584	-0.0341	-0.1510	-0.1140
A_5	-	-	-0.1067	-	-
R^2	0.9351	0.8629	0.9687	0.9614	0.9685

Table 4.29: Regression coefficients of multiple regression models predicting the compressive strength of Concrete with Zone-B of aggregates without FA

Concrete Mixes with Zone-B Aggregates					
Coefficient	f_{c28}	f_{c56}	f_{c91}	$f_{c91,28}$	$f_{c91,56}$
A_0	4.9848	9.4214	2.9227	2.9093	3.7531

A_1	-2.6623	-2.0157	-2.6871	-2.6907	-2.5446
A_2	0.5947	0.4259	0.5915	0.5923	0.5726
A_3	0.1152	0.1029	0.3159	0.3161	0.2954
A_4	-	-0.0037	-0.2122	-0.2176	-0.1716
A_5	-	-	-0.0050	-	-
R^2	0.9518	0.9088	0.9502	0.9502	0.9493

Table 4.30: Regression coefficients of multiple regression models predicting the compressive strength of Concrete with Zone-B of aggregates with 15% FA

Concrete Mixes with Zone-B Aggregates					
Coefficient	f_{c28}	f_{c56}	f_{c91}	$f_{c91,28}$	$f_{c91,56}$
A_0	10.4992	3.5841	20.3021	23.0852	18.0747
A_1	-1.8512	-2.8598	-1.1693	-0.9357	-1.0801
A_2	0.1232	0.6016	0.0316	-0.1184	-0.0796
A_3	-0.1086	0.3275	-0.0189	-0.1252	-0.0667
A_4	-	0.0263	0.2148	-0.0935	-0.1308
A_5	-	-	-0.2478	-	-
R^2	0.9771	0.8969	0.9892	0.9749	0.9844

Table 4.31: Regression coefficients of multiple regression models predicting the compressive strength of Concrete with Zone-C of aggregates without FA

Concrete Mixes with Zone-C Aggregates					
Coefficient	f_{c28}	f_{c56}	f_{c91}	$f_{c91,28}$	$f_{c91,56}$
A_0	0.2970	4.152	1.5065	2.4590	2.1313
A_1	-6.0311	-3.4970	-4.7240	-4.1082	-4.3820
A_2	2.3580	1.3572	1.8668	1.5940	1.7299
A_3	0.1728	-0.0013	0.1001	0.0914	0.0786
A_4	-	0.1490	-0.156	0.1438	0.1874
A_5	-	-	0.3275	-	-
R^2	0.9572	0.9179	0.9787	0.9753	0.9779

Table 4.32: Regression coefficients of multiple regression models predicting the compressive strength of Concrete with Zone-C of aggregates with 15% FA

Concrete Mixes with Zone-C Aggregates					
Coefficient	f_{c28}	f_{c56}	f_{c91}	$f_{c91,28}$	$f_{c91,56}$
A_0	154.1932	6.6923	19.5303	23.1378	17.4536
A_1	1.3580	-1.8834	-0.9190	-0.7240	-1.1048
A_2	-1.4933	-0.0180	-0.2415	-0.3368	-1.1340
A_3	-0.1184	0.0518	0.0397	0.0328	0.0375

A_4	-	-0.1823	-0.0590	-0.1117	-0.0986
A_5	-	-	-0.0559	-	-
R^2	0.9890	0.9286	0.9564	0.9175	0.9370

4.4.3 Results and discussions

Table 4.27 contains R^2 for concrete mixes of Zone-A aggregates without FA. It has been observed that the value of R^2 is above 90.90% for all the curing ages except f_{c28} and f_{c56} and the highest value of R^2 is 94.30%, obtained for f_{c91}. Table 4.28 contains coefficient of determination (R^2) for concrete mixes of Zone-A aggregates with 15% FA. It has been observed that the value of R^2 is above 93.51% for all the curing ages except for f_{c56} which is 86.29%. Table 4.27 and Table 4.28 also reveal that the proposed model fits better in the cases when cement contents are replaced by FA. The only exception here is the regression equation for f_{c56}.

Table 4.29 shows R^2 for concrete mixes of Zone-B aggregates without FA. It has been observed that it is higher than 94.93% for all the curing ages except f_{c56}. Table 4.30 contains R^2 for concrete mixes of Zone-B aggregates with 15% FA. It has been observed that the value of R^2 is higher than 97.71% in all the cases except $f_{c56,}$ *i.e.,* 89.69%. The values of R^2 obtained for FA concrete mixes are again found to be higher than the values of R^2 for Zone-B aggregates without FA. On comparing Table 4.29 and Table 4.30, one can observe that prediction model for compressive strength of concrete with Zone-B aggregates with 15% FA has higher reliability as compared to concrete mixes without FA.

The value of R^2 for concrete mixes of Zone-C aggregates without FA are presented in Table 4.31. It has been observed that the values of R^2 for all curing ages are above 91.79%. Table 4.32 shows R^2 for concrete mixes of Zone-C aggregates with 15% FA. It can be seen here that all curing ages have value of R^2 higher than 91.75% and it is 98.90% for f_{c28} which is considered as a very good fit, indicating that the model is best suited for strength prediction at early age. On comparing the results of Table 4.31 and Table 4.32, it has been observed that the compressive strength of concrete with Zone-C aggregates without FA can be predicted better than concretes where cement is replaced with FA except $f_{c28.}$

4.4.4 Conclusions

The work presented in the preceding section includes multiple regression models for predicting compressive strength of concrete with three different aggregate zones, *i.e*., Zone-A, Zone-B and Zone-C with and without FA. It can be observed that, although the value of R^2 is higher than 80.00% for all cases which is generally considered a good fit, it is lowest for f_{c56}. It has also been observed that for Zone-A aggregates, the values of R^2 are higher, when cement is replaced with 15% of FA than the value of R^2 without any replacement of cement with an exception for f_{c56}. The prediction of compressive strength of concrete for Zone-B aggregates is also on the same trend line as for Zone-A aggregates. But for Zone-C aggregates, the values of R^2 are higher for the situations when cement is not replaced by FA.

4.5 Development of ridge regression models

4.5.1 Dataset used

Dataset used for ridge regression models is the same as the dataset used for regression models for the prediction of compressive strength of concrete for varying workability/effect of workability on compressive strength (Section 4.3.1), available in Tables 4.3 - 4.6.

4.5.2 Modeling of data

For ridge regression analysis, data models of Section 4.3.2 (Model-2 and Model-3) have been used for the present study. The main objective of this study is to analyze the effect of variation in data on the regression coefficients of two models. We have considered six cases here. These are, *Case*-1: when complete set of data (number of tuples = 14) is considered; *Case*-2: when number of tuples in dataset is decreased (number of tuples is reduced from 14 to 8); *Case*-3: when the ridge parameter (k) is 0.3 and complete set of data is considered; *Case*-4: when the ridge parameter (k) is 0.3 and number of tuples are reduced from 14 to 8; *Case*-5: when the ridge parameter (k) is 0.66 and complete set of data (number of tuples = 14) is considered; *Case*-6: when a ridge parameter (k) is 0.66 and number of tuples are reduced from 14 to 8.

4.5.3 Results and discussions

Tables 4.9 - 4.12 in Section 4.3.2 present the results of two models, Model-2 and Model-3 for *Case*-1. Tables 4.33 - 4.36 contain the results of Model-2 and Model-3 for *Case*-2. The results of *Case*-3 and *Case*-4 for Model-2 and Model-3 are presented in Tables 4.37 - 4.40 and Tables 4.41 –

4.44 respectively, while Tables 4.45 - 4.48 and Tables 4.49 - 4.52 present results for *Case*-5 and *Case*-6, respectively.

Table 4.33: Regression coefficients for Model-2 with medium workability, as per *Case*-2

Concrete Mixes with Medium Workability (Model-2, *Case*-2)					
Coefficient	f_{c28}	f_{c56}	f_{c91}	$f_{c91,28}$	$f_{c91,56}$
A_0	1.0000	1.0000	1.0000	1.0000	1.0000
A_1	-4.8487	-6.5630	-5.4625	-4.8765	-3.8238
A_2	1.4455	-8.3069	-3.9954	-4.4318	2.9971
A_3	-0.1432	0.1026	0.0574	0.0242	-0.0241
A_4	0.5758	13.8050	7.8043	7.8227	-2.0846
A_5	-	-0.4944	-1.2849	-0.6712	-0.2257
A_6	-	-	0.9064	-	-

Table 4.34: Regression coefficients for Model-2 with high workability, as per *Case*-2

Concrete Mixes with High Workability (Model-2, *Case*-2)					
Coefficient	f_{c28}	f_{c56}	f_{c91}	$f_{c91,28}$	$f_{c91,56}$
A_0	1.0001	0.9999	1.0000	1.0000	1.0000
A_1	17.2150	19.7950	21.7450	20.4670	20.8940
A_2	-53.9500	-62.2410	-65.8760	-62.8130	-64.0360
A_3	54.8520	63.9580	66.9140	64.0600	65.2800
A_4	0	-0.0125	-0.0091	-0.0121	-0.0118
A_5	-	0.4086	-0.2188	0.1189	0.1467
A_6	-	-	0.2823	-	-

Table 4.35: Regression coefficients for Model-3 with medium workability, as per *Case*-2

Concrete Mixes with Medium Workability (Model-3, *Case*-2)					
Coefficient	f_{c28}	f_{c56}	f_{c91}	$f_{c91,28}$	$f_{c91,56}$
B_0	0.0005	0.0000	0.0000	0.0000	0.0000
B_1	2.7178	4.6605	5.0901	4.1002	4.7504
B_2	0.0028	0.0036	0.0033	0.0028	0.0031
B_3	0.5134	0.4118	0.3999	0.5693	0.4655
B_4	2.6187	2.2912	2.3583	2.4017	2.3618
B_5	-	-0.1039	0.0463	-0.0442	-0.0978
B_6	-	-	-0.1763	-	-

Table 4.36: Regression coefficients for Model-3 with high workability, as per *Case*-2

Concrete Mixes with High Workability (Model-3, *Case*-2)					
High workability concrete mixes					
Coefficient	f_{c28}	f_{c56}	f_{c91}	$f_{c91,28}$	$f_{c91,56}$
B_0	0.0000	0.0000	0.0000	0.0000	0.0000
B_1	3.0099	5.6218	4.6651	5.1006	4.5541
B_2	0.0008	0.0054	0.0025	0.0039	0.0023
B_3	6.8585	1.3285	5.1375	3.0826	5.4525
B_4	-0.3597	0.1125	-0.1708	0.0005	-0.1973
B_5	-	-0.1685	-0.0155	-0.1209	-0.1016
B_6	-	-	-0.0923	-	-

Table 4.37: Regression coefficients for Model-2 with medium workability, as per *Case*-3

Concrete Mixes with Medium Workability (Model-2, *Case*-3)					
Coefficient	f_{c28}	f_{c56}	f_{c91}	$f_{c91,28}$	$f_{c91,56}$
A_0	44.0810	49.2430	52.1020	52.1020	52.1020
A_1	-3.8508	-2.7828	-2.9878	-2.9547	-2.6902
A_2	-3.2109	-3.2771	-3.3937	-3.4196	-3.3565
A_3	0.1547	0.0621	-0.4129	-0.4747	-0.4508
A_4	1.1733	1.3371	0.8406	0.8864	0.8645
A_5	-	1.0851	-0.9602	-0.5479	-0.1812
A_6	-	-	0.4674	-	-

Table 4.38: Regression coefficients for Model-2 with high workability, as per *Case*-3

Concrete Mixes with High Workability (Model-2, *Case*-3)					
Coefficient	f_{c28}	f_{c56}	f_{c91}	$f_{c91,28}$	$f_{c91,56}$
A_0	43.3200	48.7130	51.3720	51.3720	51.3720
A_1	-2.3641	-2.9040	-2.4109	-2.5299	-2.4615
A_2	-2.1803	-0.7881	-1.3381	-0.7863	-0.8582
A_3	-0.2818	-0.2553	-0.6267	-0.6645	-0.5795
A_4	0.5977	0.6078	-0.0111	-0.0304	-0.0349
A_5	-	0.6123	-0.8399	0.1827	0.4092
A_6	-	-	0.9045	-	-

Table 4.39: Regression coefficients for Model-3 with medium workability, as per *Case*-3

Concrete Mixes with Medium Workability (Model-3, *Case*-3)					
Coefficient	f_{c28}	f_{c56}	f_{c91}	$f_{c91,28}$	$f_{c91,56}$
B_0	3.7170	3.8277	3.8853	3.8853	3.8853
B_1	-0.1326	-0.0405	-0.0394	-0.0458	-0.0319

B_2	0.0327	0.0566	0.0515	0.0456	0.0524
B_3	0.0026	0.0057	0.0150	0.0186	0.0163
B_4	-0.0243	-0.0117	0.0021	-0.0019	0.0023
B_5	-	-0.0487	0.0202	-0.0089	-0.0181
B_6	-	-	-0.0328	-	-

Table 4.40: Regression coefficients for Model-3 with high workability, as per *Case*-3

Concrete Mixes with High Workability (Model-3, *Case*-3)					
Coefficient	f_{c28}	f_{c56}	f_{c91}	$f_{c91,28}$	$f_{c91,56}$
B_0	3.7056	3.8217	3.8746	3.8746	3.8746
B_1	-0.0672	0.0055	-0.0020	0.0043	-0.0121
B_2	0.0243	0.0608	0.05034	0.0503	0.0454
B_3	0.0184	0.0008	0.0044	0.0043	0.0067
B_4	-0.0066	-0.0026	0.0069	0.0063	0.0063
B_5	-	-0.0438	-0.0130	-0.0320	-0.0239
B_6	-	-	-0.0176	-	-

Table 4.41: Regression coefficients for Model-2 with medium workability, as per *Case*-4

Concrete Mixes with Medium Workability (Model-2, *Case*-4)					
Coefficient	f_{c28}	f_{c56}	f_{c91}	$f_{c91,28}$	$f_{c91,56}$
A_0	41.9010	46.8700	49.9890	49.9890	49.9890
A_1	-2.9680	-1.9905	-1.7886	-1.8818	-1.6323
A_2	-2.0513	-1.9770	-1.9357	-2.0088	-1.9308
A_3	0.1791	0.0868	-0.3565	-0.4483	-0.4293
A_4	-1.1335	-1.7302	-1.8996	-1.9563	-2.0106
A_5	-	1.0443	-1.0818	-0.5875	-0.2307
A_6	-	-	0.7726	-	-

Table 4.42: Regression coefficients for Model-2 with high workability, as per *Case*-4

Concrete Mixes with High Workability (Model-2, *Case*-4)					
Coefficient	f_{c28}	f_{c56}	f_{c91}	$f_{c91,28}$	$f_{c91,56}$
A_0	43.1160	48.2120	50.9340	50.9340	50.9340
A_1	-1.8671	-2.7211	-1.9573	-2.2250	-2.0582
A_2	-1.8050	-1.0432	-1.6043	-1.3497	-1.4156
A_3	-1.5401	-0.3552	-1.4071	-0.94863	-1.1616
A_4	0.6723	0.5599	0.4823	0.38564	0.4527
A_5	-	0.8394	-0.59482	0.52936	0.7842
A_6	-	-	1.1055	-	-

Table 4.43: Regression coefficients for Model-3 with medium workability, as per *Case*-4

Concrete Mixes with Medium Workability (Model-3, *Case*-4)					
Coefficient	f_{c28}	f_{c56}	f_{c91}	$f_{c91,28}$	$f_{c91,56}$
B_0	3.6262	3.7354	3.8006	3.8006	3.8006
B_1	-0.1194	-0.0345	-0.0289	-0.0355	-0.0258
B_2	0.0225	0.0489	0.0398	0.0376	0.0406
B_3	0.0015	0.0067	0.0139	0.0170	0.0153
B_4	0.0023	0.0171	0.0222	0.0246	0.0240
B_5	-	-0.0458	0.0149	-0.0037	-0.0136
B_6	-	-	-0.0278	-	-

Table 4.44: Regression coefficients for Model-3 with high workability, as per *Case*-4

Concrete Mixes with High Workability (Model-3, *Case*-4)					
Coefficient	f_{c28}	f_{c56}	f_{c91}	$f_{c91,28}$	$f_{c91,56}$
B_0	3.6580	3.7669	3.8203	3.8203	3.8203
B_1	-0.0612	-0.0194	-0.0188	-0.0196	-0.0234
B_2	0.0164	0.0604	0.0425	0.0461	0.0377
B_3	0.0415	-0.0167	0.0166	0.0038	0.0230
B_4	-0.0134	0.0022	-0.0015	0.0019	-0.0033
B_5	-	-0.0403	-0.01012	-0.0303	-0.0251
B_6	-	-	-0.0201	-	-

Table 4.45: Regression coefficients for Model-2 with medium workability, as per *Case*-5

Concrete Mixes with Medium Workability (Model-2, *Case*-5)					
Coefficient	f_{c28}	f_{c56}	f_{c91}	$f_{c91,28}$	$f_{c91,56}$
A_0	42.9990	48.0340	50.8230	50.8230	50.8230
A_1	-3.5058	-2.7004	-2.7991	-2.8006	-2.6704
A_2	-3.2652	-3.0027	-3.0987	-3.1140	-3.0370
A_3	-0.0261	-0.0756	-0.5105	-0.5566	-0.5211
A_4	0.9637	1.1377	0.6712	0.7057	0.6851
A_5	-	1.2122	-0.5252	-0.2583	-0.0018
A_6	-	-	0.3323	-	-

Table 4.46: Regression coefficients for Model-2 with high workability, as per *Case*-5

Concrete Mixes with High Workability (Model-2, *Case*-5)					
Coefficient	f_{c28}	f_{c56}	f_{c91}	$f_{c91,28}$	$f_{c91,56}$
A_0	42.2560	47.516	50.1110	50.1110	50.1110
A_1	-2.2389	-2.3129	-2.0686	-2.0504	-2.0232
A_2	-2.0710	-1.1937	-1.4000	-1.1456	-1.2006

A_3	-0.4253	-0.3425	-0.6487	-0.6738	-0.6067
A_4	0.3868	0.4580	-0.0983	-0.0874	-0.0982
A_5	-	0.6361	-0.5045	0.2348	0.4193
A_6	-	-	0.7081	-	-

Table 4.47: Regression coefficients for Model-3 with medium workability, as per *Case*-5

Concrete Mixes with Medium Workability (Model-3, *Case*-5)					
Coefficient	f_{c28}	f_{c56}	f_{c91}	$f_{c91,28}$	$f_{c91,56}$
B_0	3.6257	3.7337	3.7899	3.7899	3.7899
B_1	-0.0999	-0.0383	-0.0349	-0.03891	-0.0319
B_2	0.0463	0.0536	0.0490	0.0457	0.0490
B_3	0.0115	0.0087	0.0171	0.0202	0.0174
B_4	-0.0105	-0.0078	0.0044	0.0017	0.0042
B_5	-	-0.0473	0.0104	-0.0110	0.0181
B_6	-	-	-0.0255	-	-

Table 4.48: Regression coefficients for Model-3 with high workability, as per *Case*-5

Concrete Mixes with High Workability (Model-3, *Case*-5)					
Coefficient	f_{c28}	f_{c56}	f_{c91}	$f_{c91,28}$	$f_{c91,56}$
B_0	3.6146	3.7279	3.7794	3.7794	3.7794
B_1	-0.0589	-0.0073	-0.0096	-0.0065	-0.0161
B_2	0.0278	0.0501	0.0434	0.0417	0.0406
B_3	0.0204	0.0037	0.0057	0.0061	0.0077
B_4	-0.0033	-0.0035	0.0059	0.0051	0.0059
B_5	-	-0.0370	-0.0107	-0.0268	-0.0221
B_6	-	-	-0.0167	-	-

Table 4.49: Regression coefficients for Model-2 with medium workability, as per *Case*-6

Concrete Mixes with Medium Workability (Model-2, *Case*-6)					
Coefficient	f_{c28}	f_{c56}	f_{c91}	$f_{c91,28}$	$f_{c91,56}$
A_0	40.1590	44.9210	47.9110	47.9110	47.9110
A_1	-2.4138	-1.8632	-1.6823	-1.7580	-1.5986
A_2	-2.0028	-1.8808	-1.8141	-1.8753	-1.7840
A_3	0.0829	0.0259	-0.3808	-0.4486	-0.3997
A_4	-1.5857	-1.7842	-1.8461	-1.8957	-1.8573
A_5	-	1.0987	-0.6294	-0.3161	0.0659
A_6	-	-	0.55054	-	-

Table 4.50: Regression coefficients for Model-2 with high workability, as per *Case*-6

Concrete Mixes with High Workability (Model-2, *Case*-6)					
Coefficient	f_{c28}	f_{c56}	f_{c91}	$f_{c91,28}$	$f_{c91,56}$
A_0	41.3230	46.2080	48.8160	48.8160	48.8160
A_1	-1.7910	-2.0553	-1.7977	-1.8590	-1.7962
A_2	-1.6922	-1.1599	-1.4784	-1.3586	-1.4217
A_3	-1.5382	-0.8024	-1.3388	-1.1419	-1.2721
A_4	0.5201	0.4935	0.3497	0.2986	0.3459
A_5	-	0.8209	-0.2312	0.5672	0.7712
A_6	-	-	0.8897	-	-

Table 4.51: Regression coefficients for Model-3 with medium workability, as per *Case*-6

Concrete Mixes with Medium Workability (Model-3, *Case*-6)					
Coefficient	f_{c28}	f_{c56}	f_{c91}	$f_{c91,28}$	$f_{c91,56}$
B_0	3.4755	3.5801	3.6426	3.6426	3.6426
B_1	-0.0831	-0.0339	-0.0273	-0.0313	-0.0262
B_2	0.0304	0.0432	0.0366	0.0358	0.0369
B_3	0.0069	0.0074	0.01448	0.0168	0.0148
B_4	0.0252	0.0233	0.0254	0.0273	0.0255
B_5	-	-0.0429	0.0057	-0.0061	-0.0143
B_6	-	-	-0.0189	-	-

Table 4.52: Regression coefficients for Model-3 with high workability, as per *Case*-6

Concrete Mixes with High Workability (Model-3, *Case*-6)					
Coefficient	f_{c28}	f_{c56}	f_{c91}	$f_{c91,28}$	$f_{c91,56}$
B_0	3.5059	3.6104	3.6615	3.6615	3.6615
B_1	-0.0504	-0.0218	-0.0215	-0.0220	-0.0220
B_2	0.0246	0.0461	0.0373	0.0377	0.0377
B_3	0.0383	0.0011	0.0176	0.0119	0.0119
B_4	-0.0076	-0.0019	-0.0011	0.0004	0.0004
B_5	-	-0.0338	-0.0106	-0.0266	-0.0266
B_6	-	-	-0.0185	-	-

Figs. 4.1 - 4.10 present the effect of variation in data on regression coefficients for *Case*-1 and *Case*-2 when Model-2 is used. Similar graphs for Model-3 are given in Figs. 4.11 - 4.20. One can observe from these figures that there is significant change in the value of coefficients, when data size is reduced from 14 tuples to 8 only. This statement is true for both models, used either with medium workability or with high workability. *Case*-3 and *Case*-4, for Model-2 and Model-3, are

represented in Figs. 4.21 - 4.30 and Figs. 4.31 - 4.40, respectively. It has been observed that most of the regression coefficients are stable or there is a minor change between the coefficients of *Case*-3 and *Case*-4. These trends have also been seen for *Case*-5 and *Case*-6 for their respective models *i.e.,* Model-2 and Model-3.

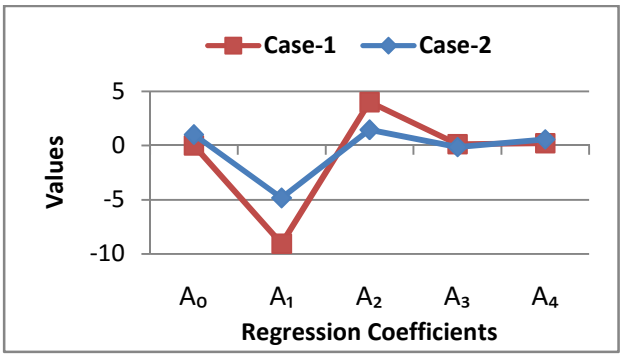

Fig. 4.1: Regression coefficients for medium workability for *Case*-1 and *Case*-2, of Model-2 when f_{c28} is the dependent variable

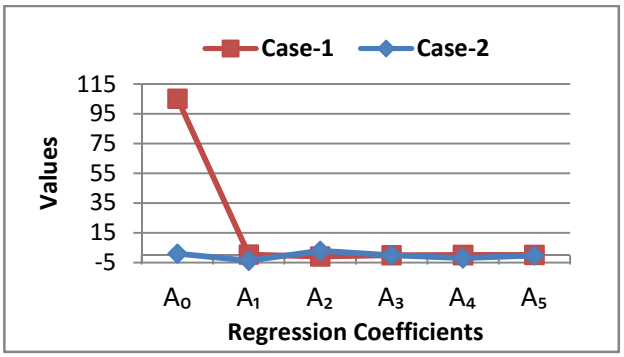

Fig. 4.5: Regression coefficients for medium workability for *Case*-1 and *Case*-2, of Model-2 when $f_{c91,56}$ is the dependent variable

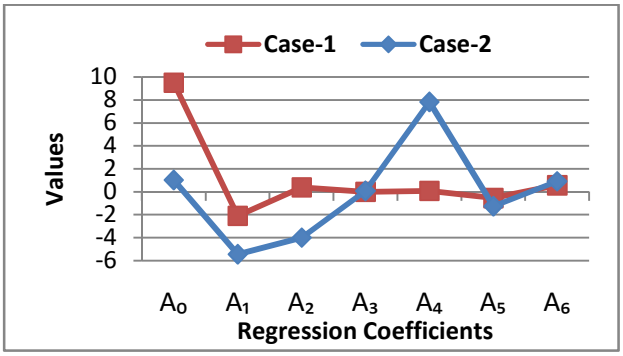

Fig. 4.3: Regression coefficients for medium workability for *Case*-1 and *Case*-2, of Model-2 when f_{c91} is the dependent variable

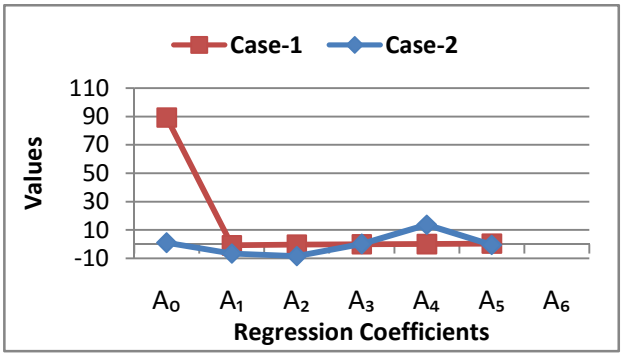

Fig. 4.2: Regression coefficients for medium workability for *Case*-1 and *Case*-2, of Model-2 when f_{c56} is the dependent variable

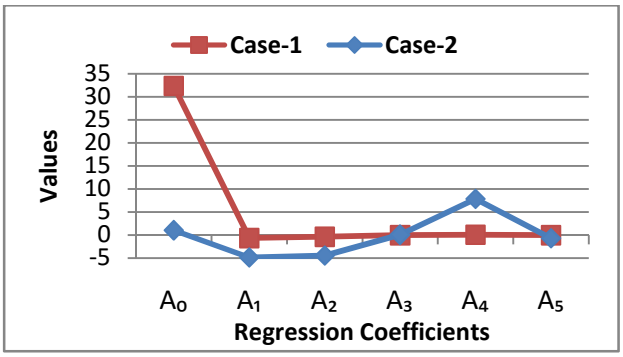

Fig. 4.4: Regression coefficients for medium workability for *Case*-1 and *Case*-2, of Model-2 when $f_{c91,28}$ is the dependent variable

Fig. 4.9: Regression coefficients for high workability for *Case*-1 and *Case*-2, of Model-2 when $f_{c91,28}$ is the dependent variable

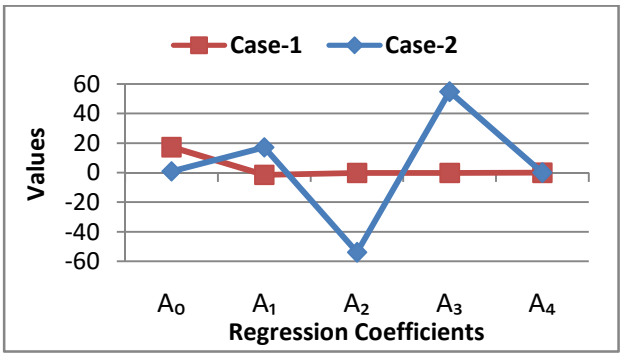

Fig. 4.6: Regression coefficients for high workability for *Case*-1 and *Case*-2, of Model-2 when f_{c28} is the dependent variable

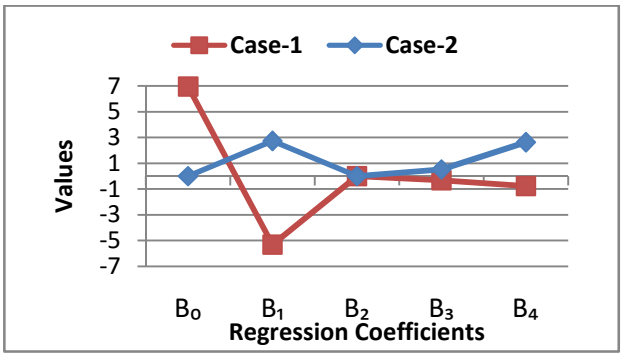

Fig. 4.11: Regression coefficients for medium workability for *Case*-1 and *Case*-2, of Model-3 when f_{c28} is the dependent variable

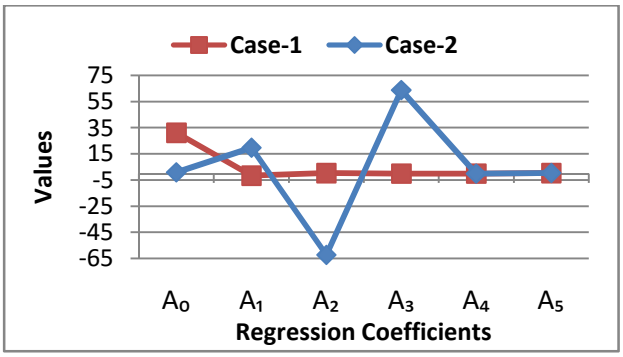

Fig. 4.7: Regression coefficients for high workability for *Case*-1 and *Case*-2, of Model-2 when f_{c56} is the dependent variable

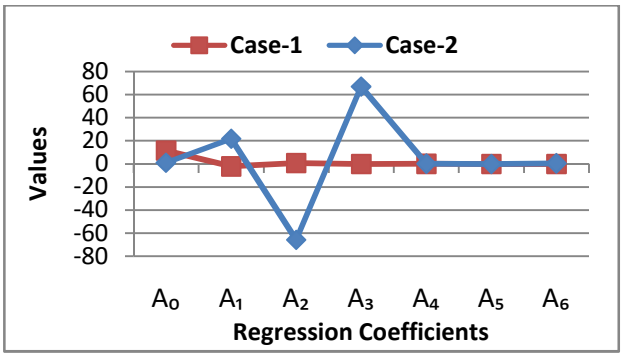

Fig. 4.8: Regression coefficients for high workability for *Case*-1 and *Case*-2, of Model-2 when f_{c91} is the dependent variable

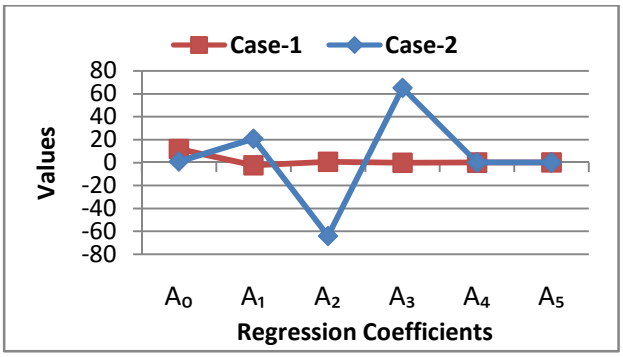

Fig. 4.10: Regression coefficients for high workability for *Case*-1 and *Case*-2, of Model-2 when $f_{c91,56}$ is the dependent variable

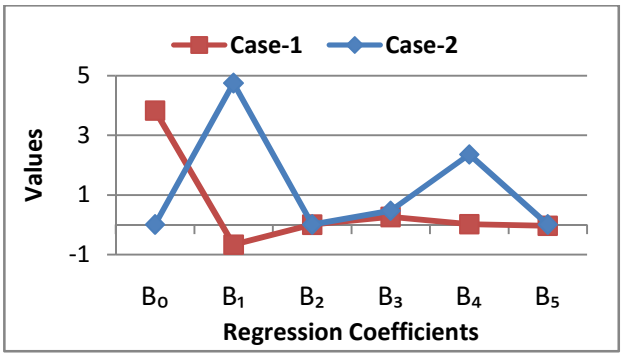

Fig. 4.15: Regression coefficients for medium workability for *Case*-1 and *Case*-2, of Model-3 when $f_{c91,56}$ is the dependent variable

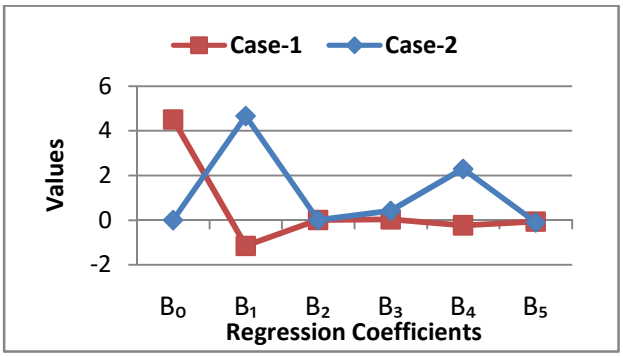

Fig. 4.12: Regression coefficients for medium workability for *Case*-1 and *Case*-2, of Model-3 when f_{c56} is the dependent variable

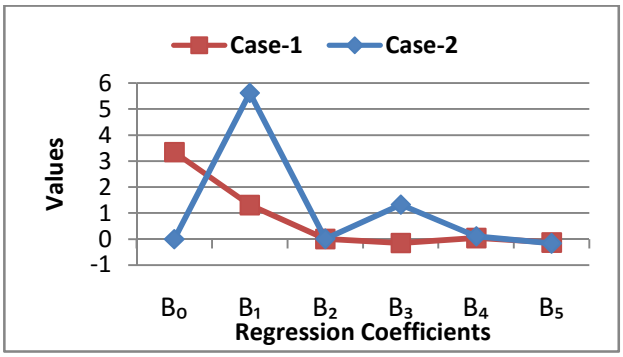

Fig. 4.17: Regression coefficients for high workability for *Case*-1 and *Case*-2, of Model-3 when f_{c56} is the dependent variable

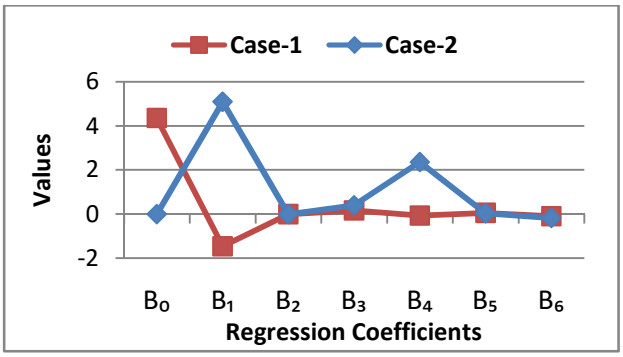

Fig. 4.13: Regression coefficients for medium workability for *Case*-1 and *Case*-2, of Model-3 when f_{c91} is the dependent variable

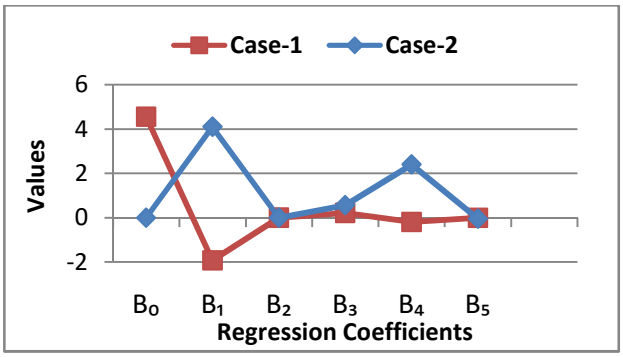

Fig. 4.14: Regression coefficients for medium workability for *Case*-1 and *Case*-2, of Model-3 when $f_{c91,28}$ is the dependent variable

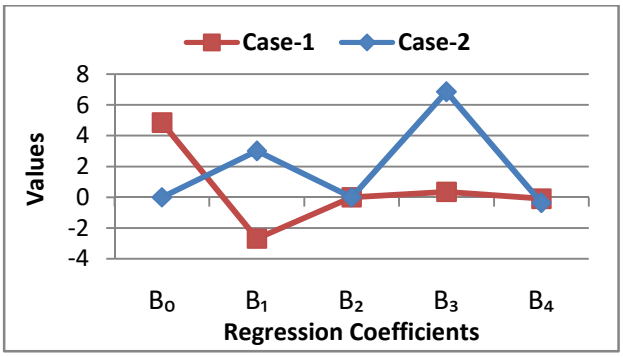

Fig. 4.16: Regression coefficients for high workability for *Case*-1 and *Case*-2, of Model-3 when f_{c28} is the dependent variable

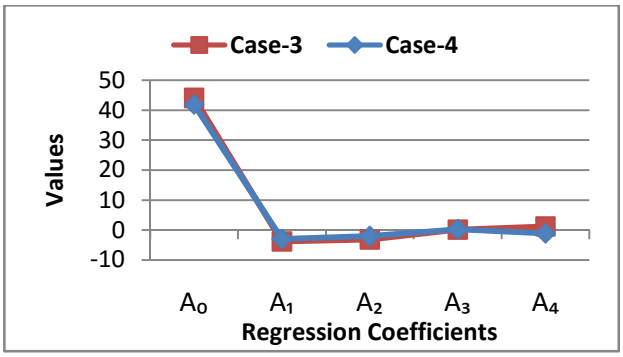

Fig. 4.21: Regression coefficients for medium workability for *Case*-3 and *Case*-4, of Model-2 when f_{c28} is the dependent variable

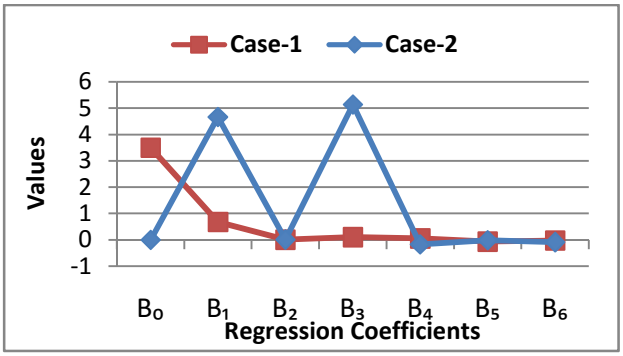

Fig. 4.18: Regression coefficients for high workability for *Case*-1 and *Case*-2, of Model-3 when f_{c91} is the dependent variable

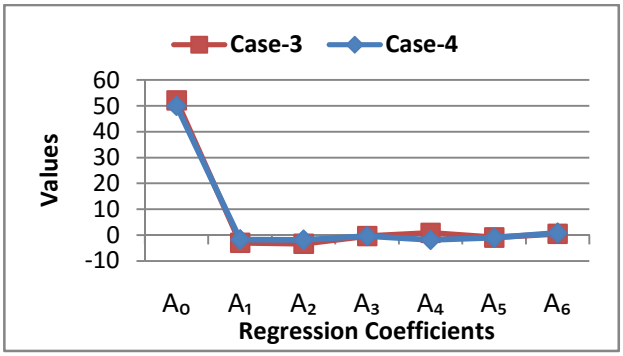

Fig. 4.23: Regression coefficients for medium workability for *Case*-3 and *Case*-4, of Model-2 when f_{c91} is the dependent variable

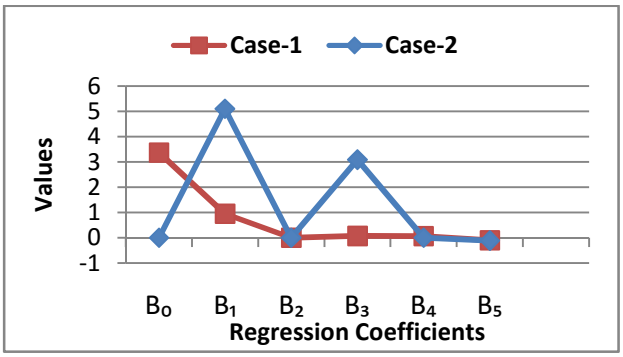

Fig. 4.19: Regression coefficients for high workability for *Case*-1 and *Case*-2, of Model-3 when $f_{c91,28}$ is the dependent variable

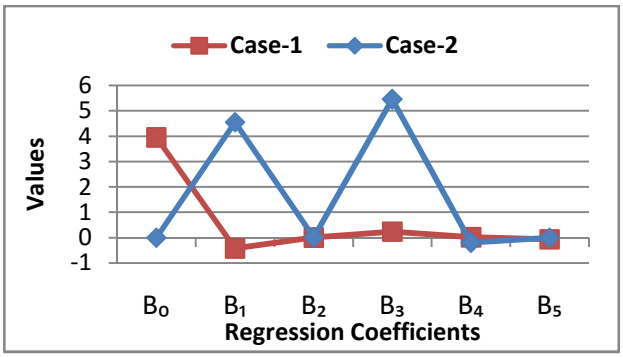

Fig. 4.20: Regression coefficients for high workability for *Case*-1 and *Case*-2, of Model-3 when $f_{c91,56}$ is the dependent variable

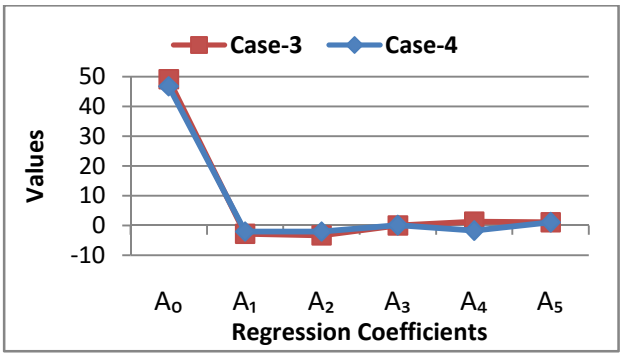

Fig. 4.22: Regression coefficients for medium workability for *Case*-3 and *Case*-4, of Model-2 when $f_{c91,56}$ is the dependent variable

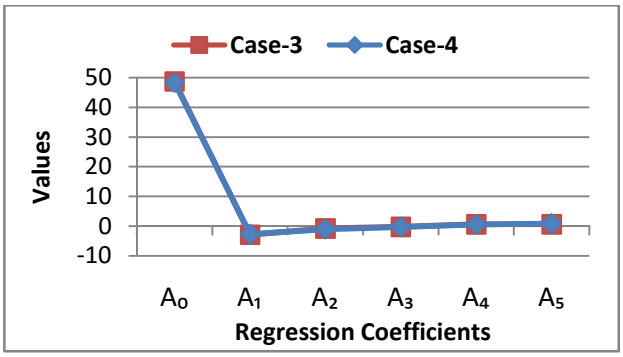

Fig. 4.27: Regression coefficients for high workability for *Case*-3 and *Case*-4, of Model-2 when f_{c56} is the dependent variable

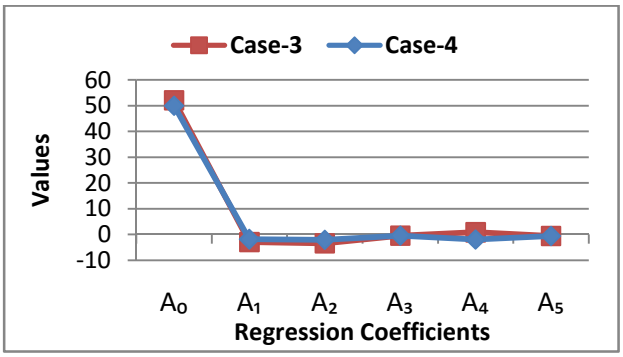

Fig. 4.24: Regression coefficients for medium workability for *Case*-3 and *Case*-4, of Model-2 when $f_{c91,28}$ is the dependent variable

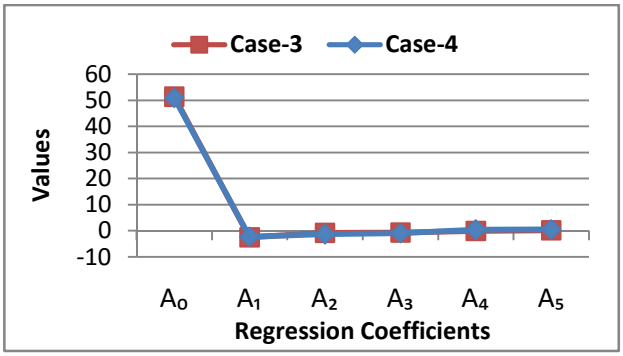

Fig. 4.29: Regression coefficients for high workability for *Case*-3 and *Case*-4, of Model-2 when $f_{c91,28}$ is the dependent variable

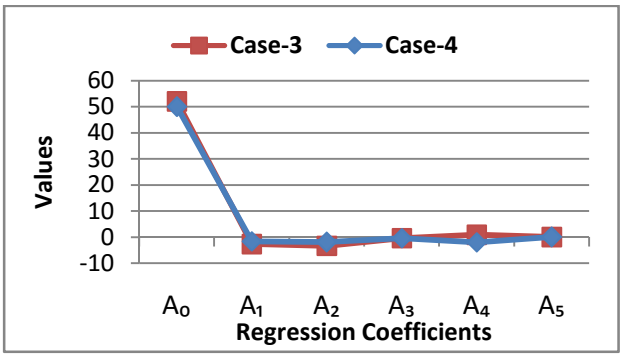

Fig. 4.25: Regression coefficients for medium workability for *Case*-3 and *Case*-4, of Model-2 when $f_{c91,56}$ is the dependent variable

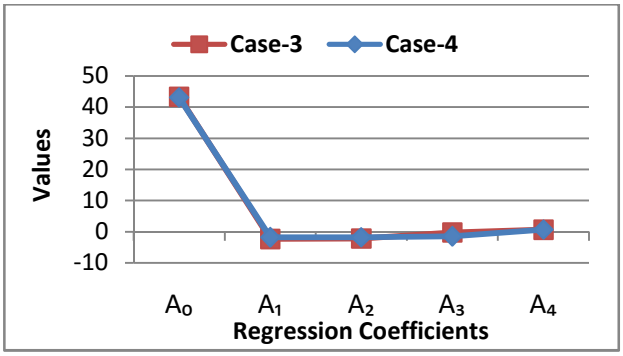

Fig. 4.26: Regression coefficients for high workability for *Case*-3 and *Case*-4, of Model-2 when f_{c28} is the dependent variable

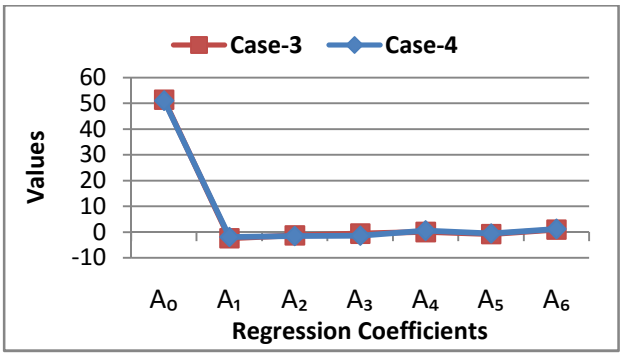

Fig. 4.28: Regression coefficients for high workability for *Case*-3 and *Case*-4, of Model-2 when f_{c91} is the dependent variable

Fig. 4.33: Regression coefficients for medium workability for *Case*-3 and *Case*-4, of Model-3 when f_{c91} is the dependent variable

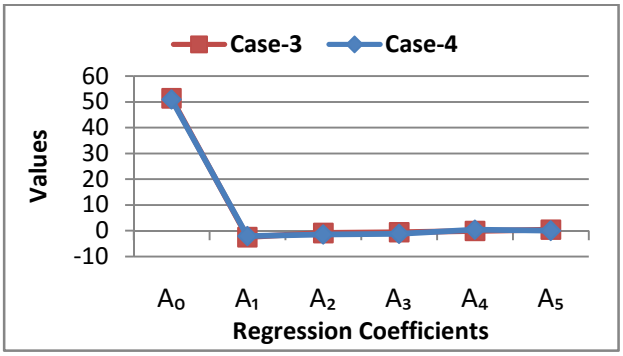

Fig. 4.30: Regression coefficients for high workability for *Case*-3 and *Case*-4, of Model-2 when $f_{c91,56}$ is the dependent variable

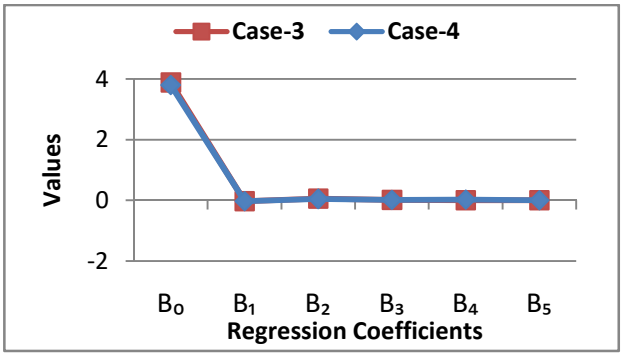

Fig. 4.35: Regression coefficients for medium workability for *Case*-3 and *Case*-4, of Model-3 when $f_{c91,56}$ is the dependent variable

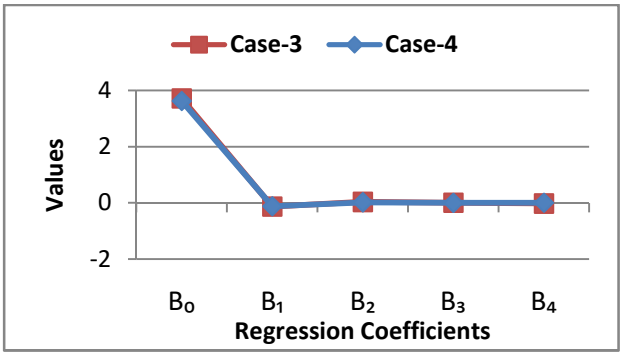

Fig. 4.31: Regression coefficients for medium workability for *Case*-3 and *Case*-4, of Model-3 when f_{c28} is the dependent variable

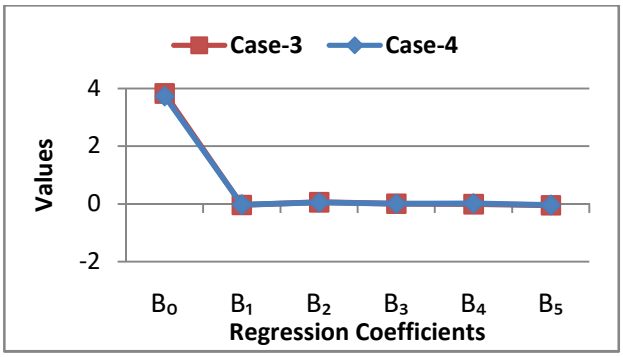

Fig. 4.32: Regression coefficients for medium workability for *Case*-3 and *Case*-4, of Model-3 when f_{c56} is the dependent variable

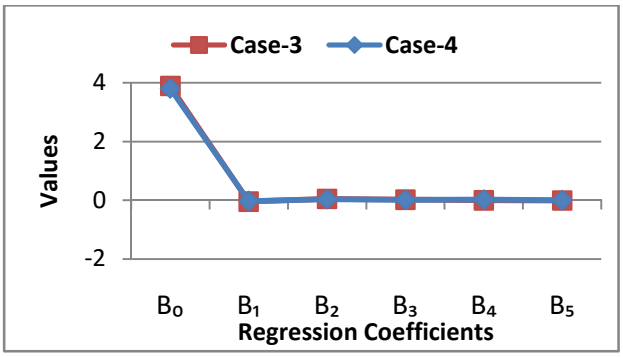

Fig. 4.34: Regression coefficients for medium workability for *Case*-3 and *Case*-4, of Model-3 when $f_{c91,28}$ is the dependent variable

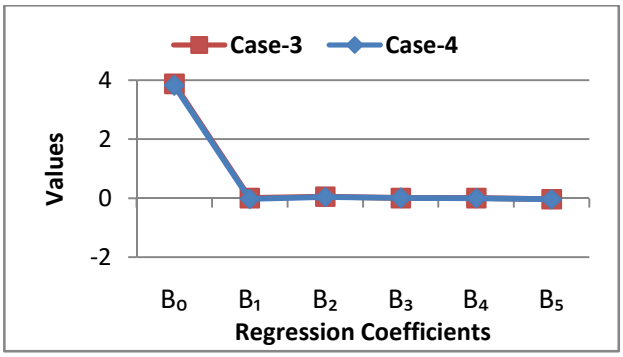

Fig. 4.39: Regression coefficients for high workability for *Case*-3 and *Case*-4, of Model-3 when $f_{c91,28}$ is the dependent variable

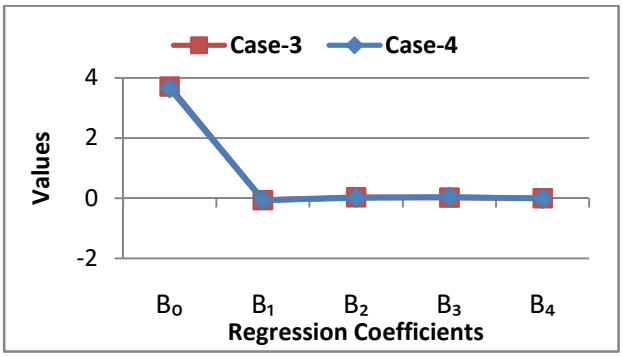

Fig. 4.36: Regression coefficients for high workability for *Case*-3 and *Case*-4, of Model-3 when f_{c28} is the dependent variable

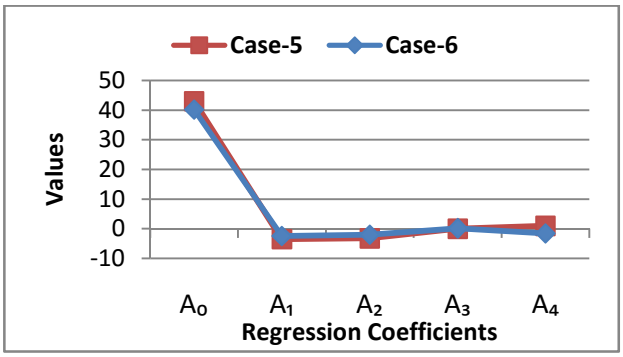

Fig. 4.41: Regression coefficients for medium workability for *Case*-5 and *Case*-6, of Model-2 when f_{c28} is the dependent variable

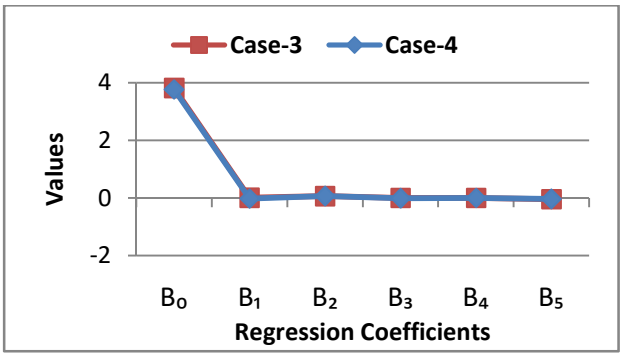

Fig. 4.37: Regression coefficients for high workability for *Case*-3 and *Case*-4, of Model-3 when f_{c56} is the dependent variable

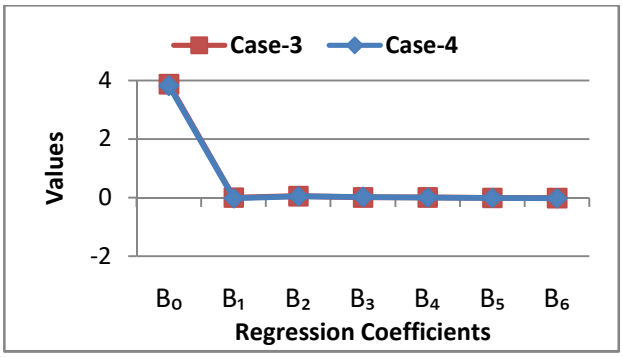

Fig. 4.38: Regression coefficients for high workability for *Case*-3 and *Case*-4, of Model-3 when f_{c91} is the dependent variable

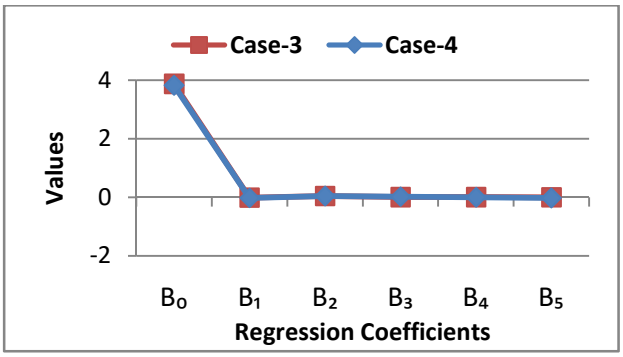

Fig. 4.40: Regression coefficients for high workability for *Case*-3 and *Case*-4, of Model-3 when $f_{c91,56}$ is the dependent variable

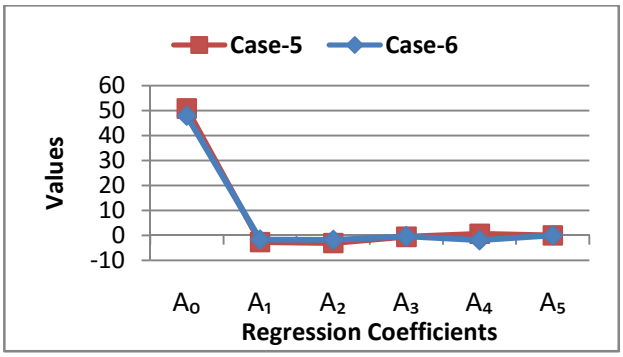

Fig. 4.45: Regression coefficients for medium workability for *Case*-5 and *Case*-6, of Model-2 when $f_{c91,56}$ is the dependent variable

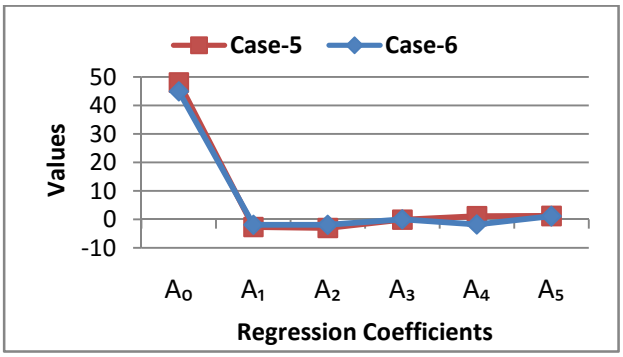

Fig. 4.42: Regression coefficients for medium workability for *Case*-5 and *Case*-6, of Model-2 when f_{c56} is the dependent variable

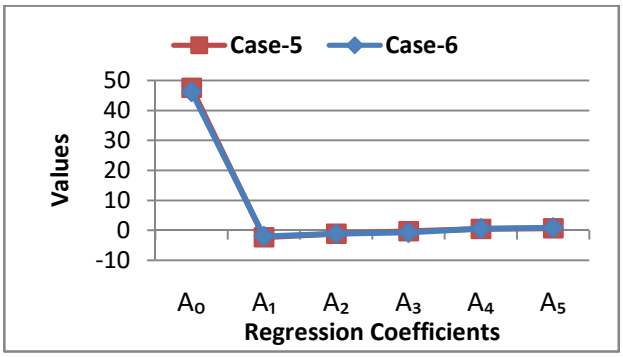

Fig. 4.47: Regression coefficients for high workability for *Case*-5 and *Case*-6, of Model-2 when f_{c56} is the dependent variable

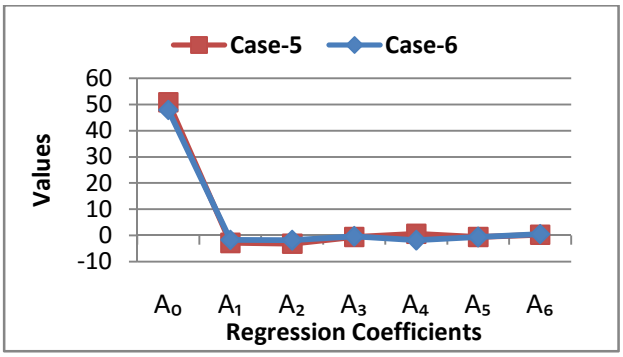

Fig. 4.43: Regression coefficients for medium workability for *Case*-5 and *Case*-6, of Model-2 when f_{c91} is the dependent variable

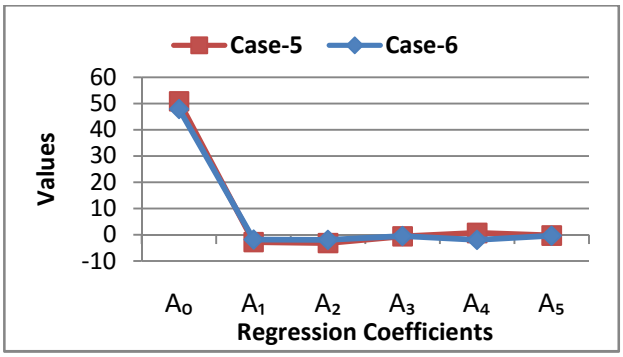

Fig. 4.44: Regression coefficients for medium workability for *Case*-5 and *Case*-6, of Model-2 when $f_{c91,28}$ is the dependent variable

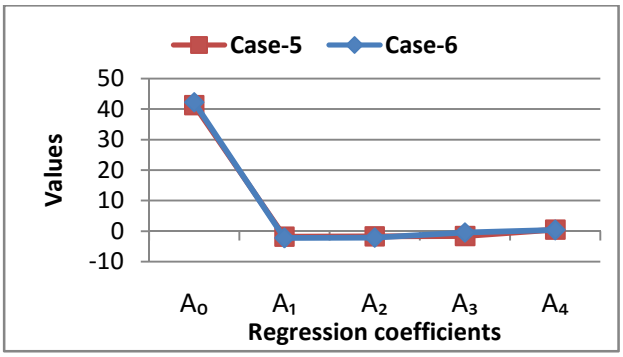

Fig. 4.46: Regression coefficients for high workability for *Case*-5 and *Case*-6, of Model-2 when f_{c28} is the dependent variable

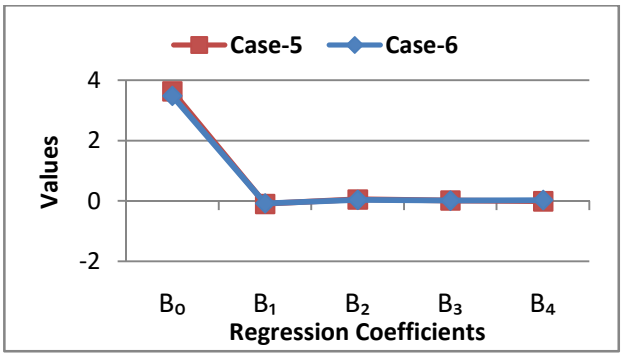

Fig. 4.51: Regression coefficients for medium workability for *Case*-5 and *Case*-6, of Model-3 when f_{c28} is the dependent variable

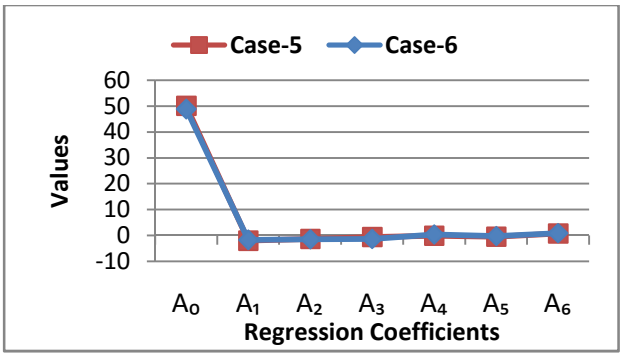

Fig. 4.48: Regression coefficients for high workability for *Case*-5 and *Case*-6, of Model-2 when f_{c91} is the dependent variable

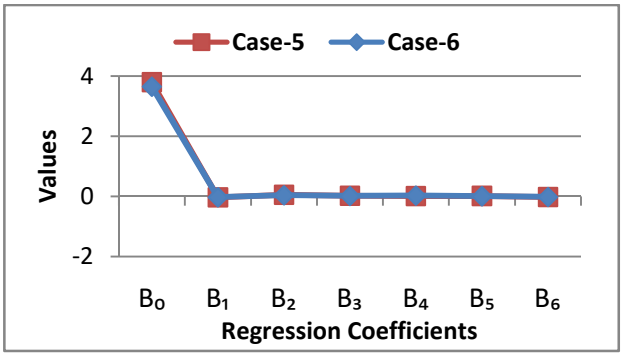

Fig. 4.53: Regression coefficients for medium workability for *Case*-5 and *Case*-6, of Model-3 when f_{c91} is the dependent variable

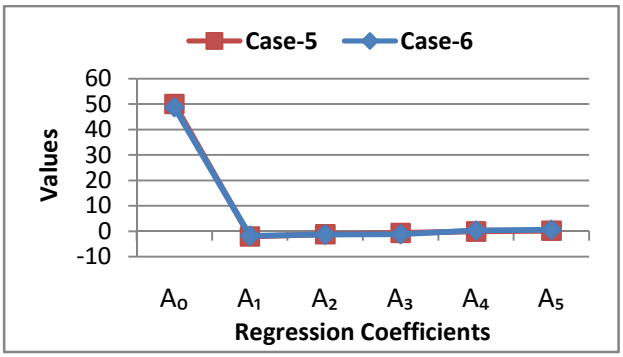

Fig. 4.49: Regression coefficients for high workability for *Case*-5 and *Case*-6, of Model-2 when $f_{c91,28}$ is the dependent variable

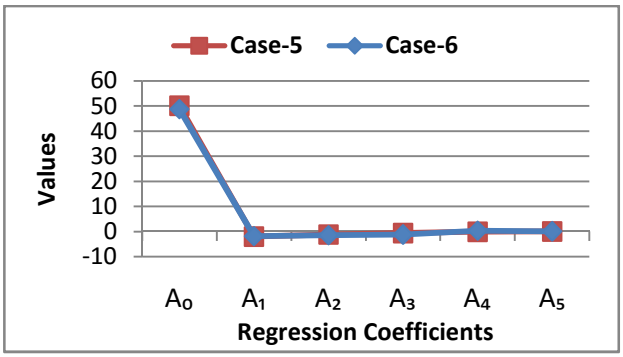

Fig. 4.50: Regression coefficients for high workability for *Case*-5 and *Case*-6, of Model-2 when $f_{c91,56}$ is the dependent variable

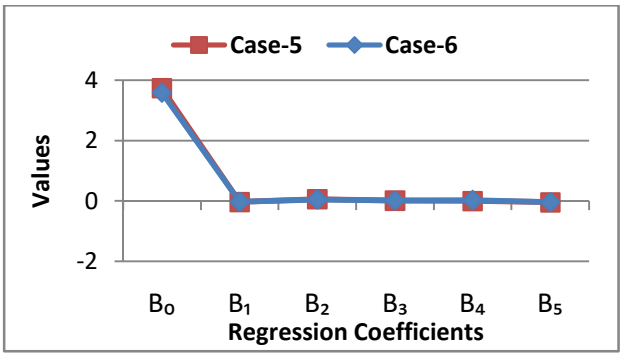

Fig. 4.52: Regression coefficients for medium workability for *Case*-5 and *Case*-6, of Model-3 when f_{c56} is the dependent variable

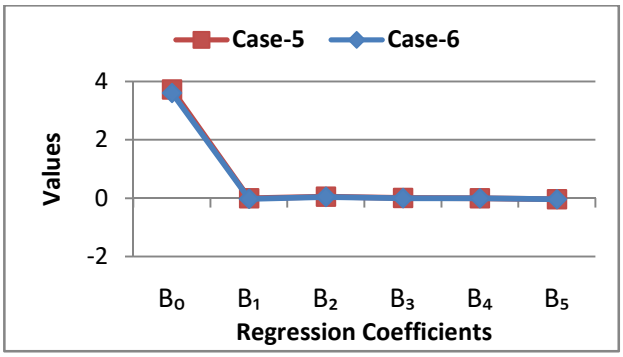

Fig. 4.57: Regression coefficients for high workability for *Case*-5 and *Case*-6, of Model-3 when f_{c56} is the dependent variable

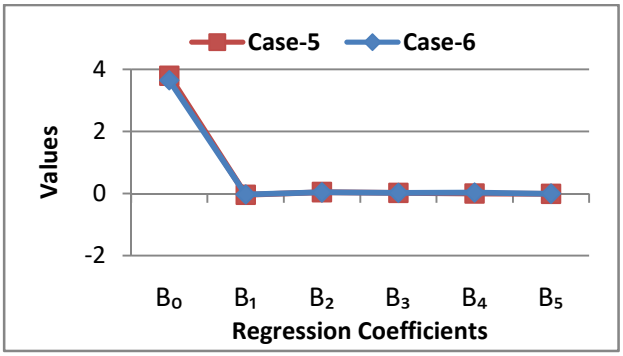

Fig. 4.54: Regression coefficients for medium workability for *Case*-5 and *Case*-6, of Model-3 when $f_{c91,28}$ is the dependent variable

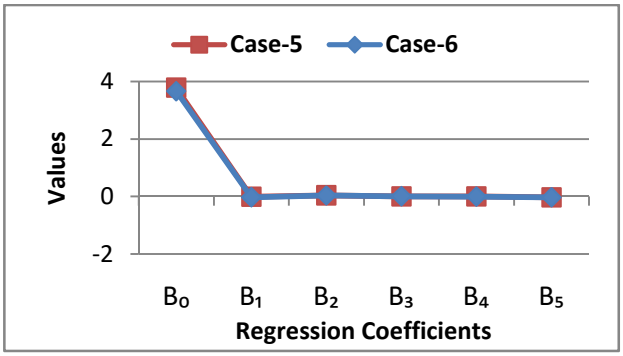

Fig. 4.59: Regression coefficients for high workability for *Case*-5 and *Case*-6, of Model-3 when $f_{c91,28}$ is the dependent variable

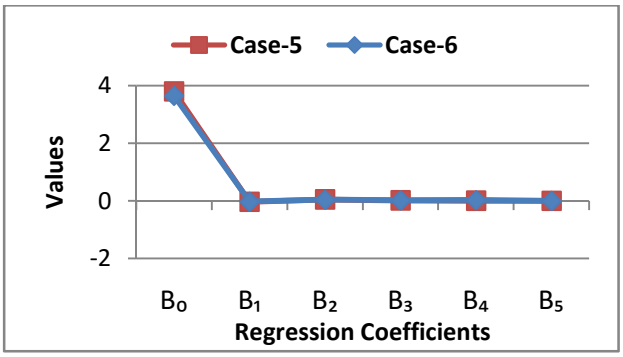

Fig. 4.55: Regression coefficients for medium workability for *Case*-5 and *Case*-6, of Model-3 when $f_{c91,56}$ is the dependent variable

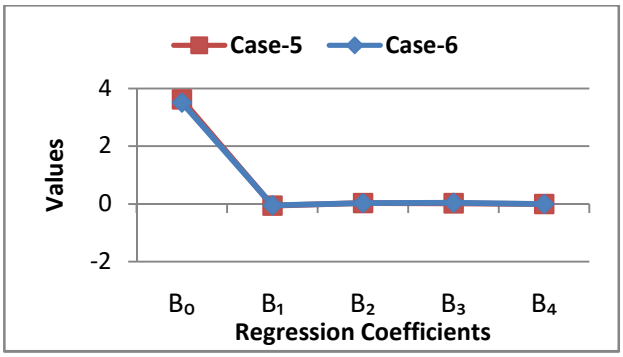

Fig. 4.56: Regression coefficients for high workability for *Case*-5 and *Case*-6, of Model-3 when f_{c28} is the dependent variable

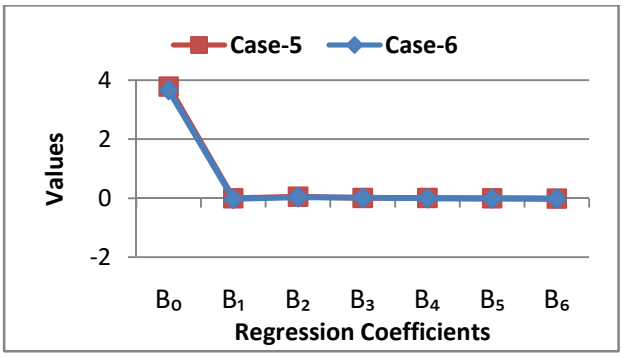

Fig. 4.58: Regression coefficients for high workability for *Case*-5 and *Case*-6, of Model-3 when f_{c91} is the dependent variable

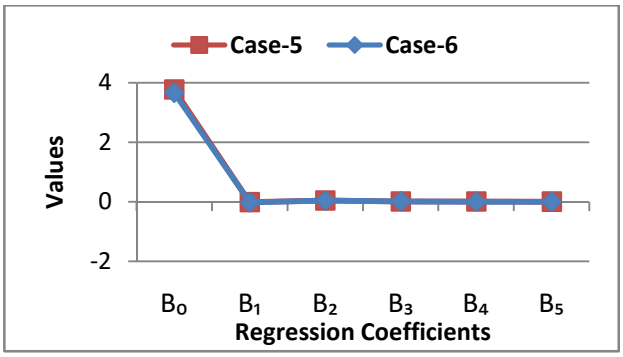

Fig. 4.60: Regression coefficients for high workability for *Case*-5 and *Case*-6, of Model-3 when $f_{c91,56}$ is the dependent variable

4.6 Conclusions

The work presented in this chapter consists of the development of regression models for predicting compressive strength of concrete at three stages of its curing. The regression models developed in Section 4.2 - 4.4 can predict compressive strength of various mixes very efficiently. However, a variation in data affects the regression coefficients to a large extent. By introducing, the ridge parameter in regression equations, we get more trustworthy and efficient predictive models for compressive strength of concrete not affected by the variations in dataset used for prediction.

CHAPTER 5

NEURAL NETWORK AND GENETIC PROGRAMMING MODELS FOR PREDICTION OF COMPRESSIVE STRENGTH OF CONCRETE AND COMPARISON OF PREDICTION MODELS

5.1 Introduction

In the previous chapters, we have developed prediction models for predicting compressive strength of concrete using conventional regression models. This is a well known fact that properties of concrete, including, compressive strength depend on a number of parameters. These parameters are affected by the non-homogeneous nature of concrete components, and also by the combined (sometimes contradictory) effects of these components. This makes the prediction task even more complex. One needs to develop models that are more accurate and are also able to predict the situations where the concrete is not able to attain designed strength, where less raw material yields higher strength. This chapter includes the development of neural network and Genetic Programming (GP) models for the prediction of compressive strength of concrete. We also carried out the comparisons of these models in this chapter. The Dataset used for development of neural network and GP models is same as used for regression models for the prediction of compressive strength of concrete (Tables 3.5 - 3.8).

5.2 Pilot neural network analysis

Neural network modeling technique has several favorable features such as competence, generalization and straightforwardness that makes it an attractive preference for modeling of intricate systems. A successful neural network model for the prediction of compressive strength of concrete requires a good conception of the effect of several internal parameters. For a feed-forward back-propagation network structure and training process, the important internal parameters include data preprocessing and presentation, initial synaptic weights, learning rate (*lr*), number of hidden layers and number of neurons in each hidden layer (*hn*), activation functions (AF) for hidden layers and the number of training epochs. In this work, a three layer feed-forward back-

propagation neural network is developed through experimental investigations of various internal parameters to predict the compressive strength of concrete. The neural network models proposed in this chapter are described in Fig. 5.1. Here, x_1 represents *w/cm*, x_2 represents *fa/cm*, x_3 represents *ca/cm*, x_4 represents f_{c28}, x_5 represents f_{c56} and x_6 represents f_{c91}. Fig. 5.2 contains a detailed architecture of the neural network model used in this work.

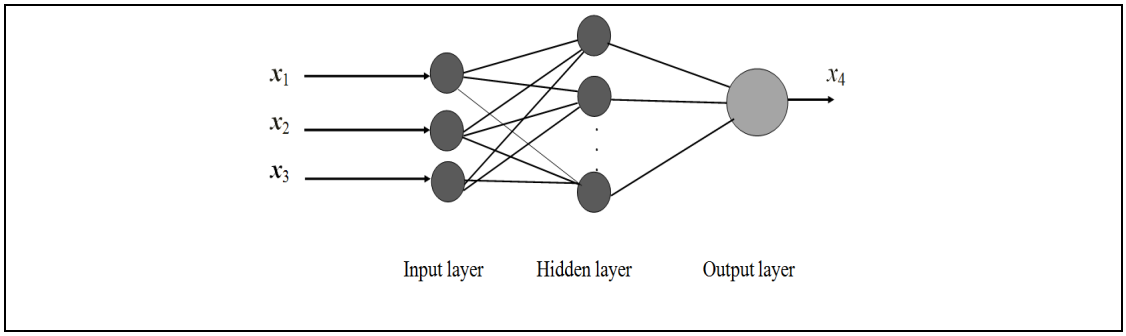

Fig. 5.1(a): Structure of neural network model with three inputs and one output

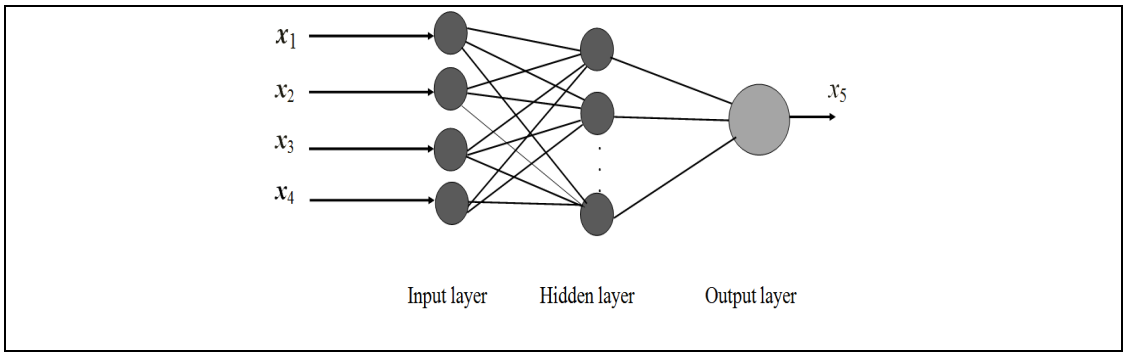

Fig. 5.1(b): Structure of neural network model with four inputs and one output

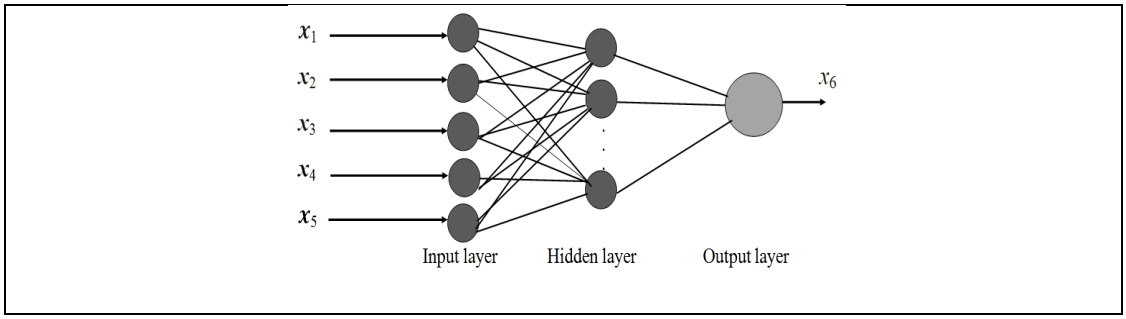

Fig. 5.1(c): Structure of neural network model with five inputs and one output

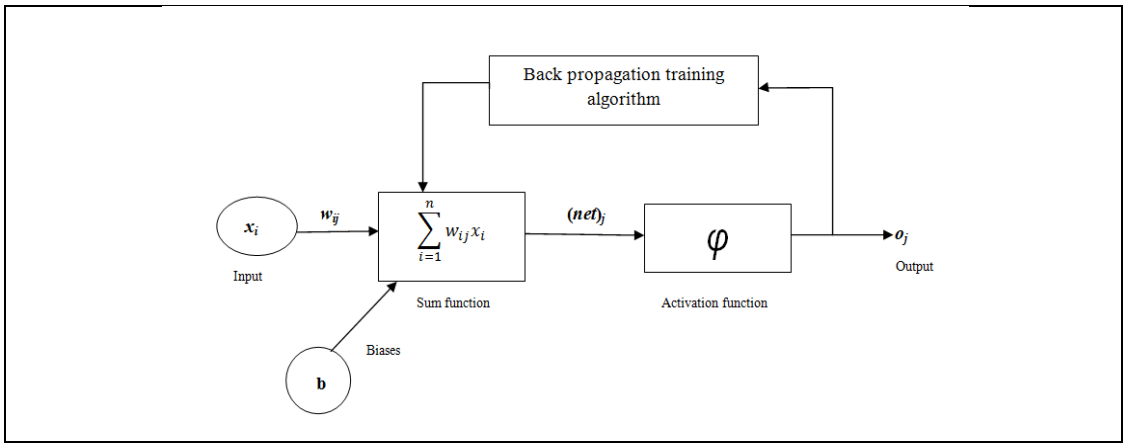

Fig. 5.2: Architecture of the developed neural network model

In Fig. 5.2, x_i is the input variable, $(net)_j$ is the weighted sum of the j^{th} neuron for the input received from the preceding layer with n neurons, w_{ij} is the weight between the j^{th} neuron of the current layer and the i^{th} neuron of the preceding layer. Output of j^{th} neuron is calculated by applying activation function (AF) on $(net)_j$.

5.2.1 Experimentation for deciding parameters of neural network models

One needs to adjust the values of parameters of neural network models during design and training phase for better prediction. However, one needs to follow the exhaustive enumeration kind of methods as theoretical foundation on the values of these parameters is not available. As such, a trial and error method is used to explore the optimum combination of parameters. The preliminary experimentation is started with certain randomly selected network architecture on the basis of the knowledge gained from literature review. We have empirically investigated 1440 neural network models for prediction of compressive strength of concrete. In this study, five important parameters, *i.e.*, *hn*, training function, AF, *lr* and *goal* have been investigated. We have considered two values (= 50, 60) for *hn*; ten types of training functions (as listed in Table 1.2); two types (= tan-sigmoid and log-sigmoid) of AF for hidden layer; three values (= 0.1, 0.01 and 0.001) of *lr*; four values (= 0.001, 0.0001, 0.00001 and 0.000001) of *goal* in this work. These values of neural network parameters have been investigated for predicting the compressive strength of concrete for three curing ages. The initial values of weights and biases are taken randomly. We have also fixed the maximum number of epochs in these experiments to 10,000.

The aim of this experimentation is to decide for the optimum values of *lr*, *goal* and *hn*. The results obtained in this experimentation are depicted in Tables 5.1 - 5.36 and graphical representations of these results are given in Figs. 5.3 - 5.38. In Tables 5.1 - 5.36, "--" represents that goal could not

be achieved in specified epochs. Following 36 cases have been considered to execute the pilot plan. ***Case*-1:** lr = 0.1, *goal* = 0.001, dataset = f_{c28} ; ***Case*-2:** lr = 0.1, *goal* = 0.001, dataset = f_{c56} ; ***Case*-3:** lr = 0.1, *goal* = 0.001, dataset = f_{c91} ; ***Case*-4:** lr = 0.1, *goal* = 0.0001, dataset = f_{c28} ; ***Case*-5:** lr = 0.1, *goal* = 0.0001, dataset = f_{c56} ; ***Case*-6:** lr = 0.1, *goal* = 0.0001, dataset = f_{c91} ; ***Case*-7:** lr = 0.1, *goal* = 0.00001, dataset = f_{c28} ; ***Case*-8:** lr = 0.1, *goal* = 0.00001, dataset = f_{c56} ; ***Case*-9:** lr = 0.1, *goal* = 0.00001, dataset = f_{c91} ; ***Case*-10:** lr = 0.1, *goal* = 0.000001, dataset = f_{c28} ; ***Case*-11:** lr = 0.1, *goal* = 0.000001, dataset = f_{c56} ; ***Case*-12:** lr = 0.1, *goal* = 0.000001, dataset = f_{c91} ; ***Case*-13:** lr = 0.01, *goal* = 0.001, dataset = f_{c28} ; ***Case*-14:** lr = 0.01, *goal* = 0.001, dataset = f_{c56} ; ***Case*-15:** lr = 0.01, *goal* = 0.001, dataset = f_{c91} ; ***Case*-16:** lr = 0.01, *goal* = 0.0001, dataset = f_{c28} ; ***Case*-17:** lr = 0.01, *goal* = 0.0001, dataset = f_{c56} ; ***Case*-18:** lr = 0.01, *goal* = 0.0001, dataset = f_{c91} ; ***Case*-19:** lr = 0.01, *goal* = 0.00001, dataset = f_{c28} ; ***Case*-20:** lr = 0.01, *goal* = 0.00001, dataset = f_{c56} ; ***Case*-21:** lr = 0.01, *goal* = 0.00001, dataset = f_{c91} ; ***Case*-22:** lr = 0.01, *goal* = 0.000001, dataset = f_{c28} ; ***Case*-23:** lr = 0.01, *goal* = 0.000001, dataset = f_{c56} ; ***Case*-24:** lr = 0.01, *goal* = 0.000001, dataset = f_{c91} ; ***Case*-25:** lr = 0.001, *goal* = 0.001, dataset = f_{c28} ; ***Case*-26:** lr = 0.001, *goal* = 0.001, dataset = f_{c56} ; ***Case*-27:** lr = 0.001, *goal* = 0.001, dataset = f_{c91} ; ***Case*-28:** lr = 0.001, *goal* = 0.0001, dataset = f_{c28} ; ***Case*-29:** lr = 0.001, *goal* = 0.0001, dataset = f_{c56} ; ***Case*-30:** lr = 0.001, *goal* = 0.0001, dataset = f_{c91} ; ***Case*-31:** lr = 0.001, *goal* = 0.00001, dataset = f_{c28} ; ***Case*-32:** lr = 0.001, *goal* = 0.00001, dataset = f_{c56} ; ***Case*-33:** lr = 0.001, *goal* = 0.00001, dataset = f_{c91} ; ***Case*-34:** lr = 0.001, *goal* = 0.000001, dataset = f_{c28} ; ***Case*-35:** lr = 0.001, *goal* = 0.000001, dataset = f_{c56} ; and ***Case*-36:** lr = 0.001, *goal* = 0.000001, dataset = f_{c91}.

Table 5.1: MSE for *Case*-1 using ten training functions and two AFs

MIX DESIGN for f_{c28}					
		MSE			
AF used in hidden layer→		**log-sigmoid**		**tan-sigmoid**	
S. No.	**Training Function↓**	***hn* = 50**	***hn* = 60**	***hn* = 50**	***hn* = 60**
1	trainbfg()	0.0109	0.0997	0.0098	**0.0084**
2	traincgp()	0.0408	0.0256	**0.0072**	0.0100
3	traincgb()	0.0727	0.0610	**0.0074**	0.0094
4	traincgf()	0.0314	0.3190	0.0100	**0.0099**
5	trainlm()	0.0059	0.0506	**0.0034**	0.0089
6	trainrp()	0.1521	0.6468	**0.0099**	0.0100
7	traingda()	0.0162	0.5432	**0.0087**	0.0174
8	trainscg()	0.0680	0.1139	0.0099	**0.0098**

9	trainoss()	**0.0040**	0.2326	0.0094	0.0097
10	trainbr()	0.0541	0.1043	0.0100	**0.0088**

Table 5.2: MSE for *Case*-2 using ten training functions and two AFs

MIX DESIGN for f_{c56}					
		MSE			
AF used in hidden layer→		**log-sigmoid**		**tan-sigmoid**	
S. No.	**Training Function↓**	***hn* = 50**	***hn* = 60**	***hn* = 50**	***hn* = 60**
1	trainbfg()	0.0070	0.0098	0.0066	**0.0066**
2	traincgp()	0.0098	**0.0086**	0.0518	0.0230
3	traincgb()	0.0083	0.0076	**0.0058**	0.4534
4	traincgf()	**0.0095**	0.0097	0.0440	0.3473
5	trainlm()	0.0066	0.0051	**0.0015**	0.0026
6	trainrp()	0.0100	**0.0099**	0.0860	0.2530
7	traingda()	0.0032	0.0030	0.0046	**0.0029**
8	trainscg()	0.1000	**0.0100**	0.0331	1.2853
9	trainoss()	**0.0074**	0.0081	0.0095	0.0228
10	trainbr()	--	--	--	--

Table 5.3 MSE for *Case*-3 using ten training functions and two AFs

MIX DESIGN for f_{c91}					
		MSE			
AF used in hidden layer→		**log-sigmoid**		**tan-sigmoid**	
S. No.	**Training Function↓**	***hn* = 50**	***hn* = 60**	***hn* = 50**	***hn* = 60**
1	trainbfg()	**0.0047**	0.0358	0.0070	0.0098
2	traincgp()	0.0141	0.0346	0.0098	**0.0086**
3	traincgb()	**0.0055**	0.0573	0.0083	0.0076
4	traincgf()	0.0168	0.0283	**0.0095**	0.0097
5	trainlm()	0.0427	0.1025	0.0066	**0.0051**
6	trainrp()	0.0941	0.0099	0.0100	**0.0099**
7	traingda()	0.0020	0.0214	0.0032	0.0030
8	trainscg()	0.0306	0.1286	**0.0100**	0.0100
9	trainoss()	0.0076	0.0489	**0.0074**	0.0081
10	trainbr()	--	--	--	--

Table 5.4: MSE for *Case*-4 using ten training functions and two AFs

MIX DESIGN for f_{c28}					
		MSE			
AF used in hidden layer→		log-sigmoid		tan-sigmoid	
S. No.	Training Function↓	*hn* = 50	*hn* = 60	*hn* = 50	*hn* = 60
1	trainbfg()	0.0136	0.0162	0.0099	**0.0076**
2	traincgp()	0.0475	0.1533	0.0095	**0.0094**
3	traincgb()	0.0171	0.1014	**0.0085**	0.0099
4	traincgf()	**0.0034**	0.2409	0.0096	0.0099
5	trainlm()	0.0033	0.0740	0.0065	**0.0025**
6	trainrp()	0.0121	0.0754	0.0099	**0.0096**
7	traingda()	**0.0019**	--	0.0032	--
8	trainscg()	0.0207	0.1031	**0.0098**	0.0100
9	trainoss()	0.0930	0.0417	**0.0095**	0.0096
10	trainbr()	0.0323	--	**0.0054**	--

Table 5.5: MSE for *Case*-5 using ten training functions and two AFs

MIX DESIGN for f_{c56}					
		MSE			
AF used in hidden layer→		log-sigmoid		tan-sigmoid	
S. No.	Training Function↓	*hn* = 50	*hn* = 60	*hn* = 50	*hn* = 60
1	trainbfg()	0.0456	0.0316	**0.0083**	0.0099
2	traincgp()	0.0119	0.1381	**0.0076**	0.0099
3	traincgb()	0.1012	0.1090	**0.0052**	0.0098
4	traincgf()	**0.0064**	0.2150	0.0089	0.0098
5	trainlm()	0.0071	0.0704	**0.0057**	0.0081
6	trainrp()	0.5380	0.1512	0.0100	**0.0099**
7	traingda()	0.0166	**0.0018**	0.0018	0.0032
8	trainscg()	0.1299	0.1758	**0.0100**	0.0100
9	trainoss()	0.5119	0.1744	**0.0086**	0.0099
10	trainbr()	--	--	--	--

Table 5.6: MSE for *Case*-6 using ten training functions and two AFs

MIX DESIGN for f_{c91}					
		MSE			
AF used in hidden layer→		log-sigmoid		tan-sigmoid	
S. No.	Training Function↓	*hn* = 50	*hn* = 60	*hn* = 50	*hn* = 60
1	trainbfg()	0.0208	0.1413	0.0080	**0.0056**
2	traincgp()	**0.0050**	0.0428	0.0071	0.0100

3	traincgb()	0.0556	0.0194	**0.0070**	0.0097
4	traincgf()	**0.0040**	0.0760	0.0089	0.0081
5	trainlm()	**0.0039**	0.0127	0.0055	0.0082
6	trainrp()	0.0224	0.1456	**0.0099**	0.0100
7	traingda()	0.0031	**0.0019**	0.0027	0.0032
8	trainscg()	0.0327	0.1451	**0.0100**	0.0100
9	trainoss()	0.0275	0.0534	**0.0083**	0.0099
10	trainbr()	--	--	--	--

Table 5.7: MSE for *Case*-7 using ten training functions and two AFs

MIX DESIGN for f_{c28}					
		MSE			
AF used in hidden layer→		**log-sigmoid**		**tan-sigmoid**	
S. No.	**Training Function↓**	***hn* = 50**	***hn* = 60**	***hn* = 50**	***hn* = 60**
1	trainbfg()	0.0122	0.3747	0.0094	**0.0092**
2	traincgp()	0.1344	0.0982	**0.0074**	0.0096
3	traincgb()	0.0164	0.0944	0.0095	**0.0093**
4	traincgf()	0.1665	0.5883	0.0099	**0.0099**
5	trainlm()	0.0106	0.3477	**0.0035**	0.0036
6	trainrp()	0.0269	0.0737	**0.0099**	0.0100
7	traingda()	0.0212	0.2939	**0.0094**	0.0097
8	trainscg()	0.0169	0.1738	**0.0100**	0.0100
9	trainoss()	0.0198	0.2162	**0.0095**	0.0100
10	trainbr()	---	---	---	---

Table 5.8: MSE for *Case*-8 using ten training functions and two AFs

MIX DESIGN for f_{c56}					
		MSE			
AF used in hidden layer→		**log-sigmoid**		**tan-sigmoid**	
S. No.	**Training Function↓**	***hn* = 50**	***hn* = 60**	***hn* = 50**	***hn* = 60**
1	trainbfg()	0.0088	0.0340	**0.0080**	0.0093
2	traincgp()	**0.0077**	0.1967	0.0098	0.0092
3	traincgb()	0.0116	0.1909	**0.0084**	0.0098
4	traincgf()	0.0105	0.1159	**0.0094**	0.0095
5	trainlm()	0.0129	0.0611	**0.0067**	0.0076
6	trainrp()	0.0756	0.2851	**0.0096**	0.0099
7	traingda()	--	**0.0149**	--	--
8	trainscg()	0.0626	0.0878	**0.0100**	0.0100
9	trainoss()	0.0123	0.0994	**0.0080**	0.0099

10	trainbr()	0.0469	0.0361	0.0099	**0.0094**

Table 5.9: MSE for *Case*-9 using ten training functions and two AFs

MIX DESIGN for f_{c91}					
		MSE			
AF used in hidden layer→		**log-sigmoid**		**tan-sigmoid**	
S. No.	**Training Function↓**	***hn* = 50**	***hn* = 60**	***hn* = 50**	***hn* = 60**
1	trainbfg()	**0.0034**	0.0141	0.0057	0.0075
2	traincgp()	**0.0049**	0.0100	0.0089	0.0094
3	traincgb()	0.0053	0.0055	**0.0049**	0.0095
4	traincgf()	**0.0045**	0.0317	0.0090	0.0095
5	trainlm()	0.0155	0.0280	0.0100	**0.0030**
6	trainrp()	0.0252	0.1411	**0.0096**	0.0099
7	traingda()	0.0054	**0.0035**	--	0.0053
8	trainscg()	**0.0097**	0.0701	0.0100	0.0100
9	trainoss()	0.0234	0.0473	0.0097	**0.0072**
10	trainbr()	--	--	--	--

Table 5.10: MSE for *Case*-10 using ten training functions and two AFs

MIX DESIGN for f_{c28}					
		MSE			
AF used in hidden layer→		**log-sigmoid**		**tan-sigmoid**	
S. No.	**Training Function↓**	***hn* = 50**	***hn* = 60**	***hn* = 50**	***hn* = 60**
1	trainbfg()	**0.0097**	0.0099	0.0634	0.0468
2	traincgp()	**0.0075**	0.0090	0.0348	0.0947
3	traincgb()	0.0099	**0.0094**	0.0105	0.0403
4	traincgf()	**0.0073**	0.0098	0.0097	0.0231
5	trainlm()	0.0047	**0.0026**	0.0034	0.0030
6	trainrp()	**0.0096**	0.0100	0.1774	0.0510
7	traingda()	----	----	----	----
8	trainscg()	**0.0100**	0.0100	0.0127	0.1683
9	trainoss()	**0.0063**	0.0100	0.0221	0.2623
10	trainbr()	**0.0097**	0.0100	0.0101	0.0660

Table 5.11: MSE for *Case*-11 using ten training functions and two AFs

MIX DESIGN for f_{c56}					
		MSE			
AF used in hidden layer→		**log-sigmoid**		**tan-sigmoid**	
S. No.	**Training Function↓**	***hn* = 50**	***hn* = 60**	***hn* = 50**	***hn* = 60**
1	trainbfg()	0.0350	0.0315	0.0098	**0.0095**
2	traincgp()	0.0426	0.0359	**0.0078**	0.0095
3	traincgb()	0.0261	0.0381	**0.0088**	0.0094
4	traincgf()	**0.0068**	0.1268	0.0075	0.0096
5	trainlm()	0.0136	0.0221	**0.0063**	0.0070
6	trainrp()	**0.0077**	0.0287	0.0099	0.0099
7	traingda()	--	--	--	--
8	trainscg()	**0.0054**	0.0330	0.0100	0.0100
9	trainoss()	0.0139	0.0331	**0.0063**	0.0086
10	trainbr()	--	0.0867	--	**0.0098**

Table 5.12: MSE for *Case*-12 using ten training functions and two AFs

MIX DESIGN for f_{c91}					
		MSE			
AF used in hidden layer→		**log-sigmoid**		**tan-sigmoid**	
S. No.	**Training Function↓**	***hn* = 50**	***hn* = 60**	***hn* = 50**	***hn* = 60**
1	trainbfg()	0.0097	**0.0092**	**0.0092**	0.0381
2	traincgp()	0.0084	0.0092	**0.0070**	0.1217
3	traincgb()	**0.0079**	0.0093	0.0264	0.1382
4	traincgf()	0.0092	0.0096	**0.0086**	0.0335
5	trainlm()	0.0045	0.0097	**0.0030**	0.0728
6	trainrp()	**0.0082**	0.0090	0.0088	0.0183
7	traingda()	**0.0020**	0.0031	0.0031	0.0057
8	trainscg()	0.0100	0.0100	**0.0071**	0.0737
9	trainoss()	0.0099	**0.0094**	0.0177	0.0453
10	trainbr()	--	--	--	--

Table 5.13: MSE for *Case*-13 using ten training functions and two AFs

MIX DESIGN for f_{c28}					
		MSE			
AF used in hidden layer→		**log-sigmoid**		**tan-sigmoid**	
S. No.	**Training Function↓**	***hn* = 50**	***hn* = 60**	***hn* = 50**	***hn* = 60**
1	trainbfg()	**0.0023**	0.0347	0.0136	0.0162
2	traincgp()	**0.0036**	0.0609	0.0275	0.0533

3	traincgb()	**0.0079**	0.051	0.0171	0.1014
4	traincgf()	0.0092	0.0563	**0.0034**	0.1409
5	trainlm()	0.0019	0.0046	**0.0012**	0.074
6	trainrp()	0.1632	0.153	**0.0121**	0.0754
7	traingda()	0.0057	0.0474	**0.0019**	0.0021
8	trainscg()	**0.0036**	0.0994	0.0207	0.1031
9	trainoss()	**0.0056**	0.0197	0.0093	0.0417
10	trainbr()	**0.0223**	0.0566	0.0323	0.1509

Table 5.14: MSE for *Case*-14 using ten training functions and two AFs

MIX DESIGN for f_{c56}					
		MSE			
AF used in hidden layer→		**log-sigmoid**		**tan-sigmoid**	
S. No.	**Training Function↓**	***hn* = 50**	***hn* = 60**	***hn* = 50**	***hn* = 60**
1	trainbfg()	0.0089	0.0368	**0.0076**	0.077
2	traincgp()	0.0252	0.1018	**0.0196**	0.1119
3	traincgb()	**0.0036**	0.0274	0.0141	0.1084
4	traincgf()	**0.0075**	0.0197	0.0129	0.116
5	trainlm()	0.0017	0.0012	**0.0011**	0.0110
6	trainrp()	**0.0085**	0.0835	0.0272	0.0346
7	traingda()	**0.0028**	--	0.0037	0.0055
8	trainscg()	**0.0094**	0.0271	0.0312	0.0781
9	trainoss()	0.0136	0.0666	**0.0114**	0.1311
10	trainbr()	0.0439	0.1337	**0.0071**	0.0769

Table 5.15: MSE for *Case*-15 using ten training functions and two AFs

MIX DESIGN for f_{c91}					
		MSE			
AF used in hidden layer→		**log-sigmoid**		**tan-sigmoid**	
S. No.	**Training Function↓**	***hn* = 50**	***hn* = 60**	***hn* = 50**	***hn* = 60**
1	trainbfg()	0.0115	0.0310	**0.0017**	0.0972
2	traincgp()	0.0120	0.0757	**0.0051**	0.0143
3	traincgb()	**0.0066**	0.0300	0.0094	0.0205
4	traincgf()	**0.0043**	0.0882	0.0100	0.1055
5	trainlm()	0.0013	0.0018	**0.0009**	0.0016
6	trainrp()	0.0132	**0.0123**	0.0261	0.0230
7	traingda()	0.0048	--	**0.0028**	0.0056
8	trainscg()	**0.0075**	0.0137	0.0088	0.0676
9	trainoss()	0.0187	0.0897	**0.0104**	0.0505

10	trainbr()	**0.0068**	0.0917	0.0101	0.0167

Table 5.16: MSE for *Case*-16 using ten training functions and two AFs

MIX DESIGN for f_{c28}					
		MSE			
AF used in hidden layer→		**log-sigmoid**		**tan-sigmoid**	
S. No.	**Training Function↓**	***hn* = 50**	***hn* = 60**	***hn* = 50**	***hn* = 60**
1	trainbfg()	**0.0139**	0.0597	0.0326	0.2341
2	traincgp()	0.1090	**0.0124**	0.0196	0.2872
3	traincgb()	**0.0287**	0.0321	0.0136	0.2415
4	traincgf()	**0.0124**	0.1327	0.1027	0.2129
5	trainlm()	0.0019	0.0018	**0.0010**	0.0013
6	trainrp()	0.1213	0.1568	0.1003	**0.0074**
7	traingda()	0.1090	0.2356	**0.0234**	0.3034
8	trainscg()	0.1065	0.3098	**0.0568**	0.0365
9	trainoss()	0.0654	0.3256	**0.0089**	0.3159
10	trainbr()	**0.0139**	0.0597	0.0326	0.2341

Table 5.17: *Case* MSE for *Case*-17 using ten training functions and two AFs

MIX DESIGN for f_{c56}					
		MSE			
AF used in hidden layer→		**log-sigmoid**		**tan-sigmoid**	
S. No.	**Training Function↓**	***hn* = 50**	***hn* = 60**	***hn* = 50**	***hn* = 60**
1	trainbfg()	0.0098	**0.0056**	0.0426	0.0112
2	traincgp()	0.1271	0.1219	**0.0312**	0.1286
3	traincgb()	0.0987	0.1545	**0.0187**	0.1122
4	traincgf()	0.0871	0.0954	0.0298	**0.0098**
5	trainlm()	**0.0014**	0.0109	0.0017	0.0201
6	trainrp()	**0.0234**	0.0981	0.0421	0.0987
7	traingda()	**0.0087**	0.1232	0.0276	0.0091
8	trainscg()	**0.0543**	0.0567	0.0256	0.0987
9	trainoss()	**0.0124**	0.0693	0.0126	0.0298
10	trainbr()	0.1569	0.0298	**0.0128**	0.0918

Table 5.18: MSE for *Case*-18 using ten training functions and two AFs

MIX DESIGN for f_{c91}					
		MSE			
AF used in hidden layer→		log-sigmoid		tan-sigmoid	
S. No.	Training Function↓	*hn* = 50	*hn* = 60	*hn* = 50	*hn* = 60
1	trainbfg()	0.2871	0.2181	**0.0028**	0.483
2	traincgp()	**0.0098**	0.1292	0.0956	0.3745
3	traincgb()	**0.0067**	0.1982	0.1368	0.4526
4	traincgf()	0.9089	0.2891	0.1938	**0.0910**
5	trainlm()	0.0018	0.0019	**0.0012**	0.0020
6	trainrp()	0.8761	0.2362	**0.0392**	0.1923
7	traingda()	0.0089	0.0091	0.0093	**0.0019**
8	trainscg()	0.2178	**0.0076**	0.1729	0.9083
9	trainoss()	0.8129	**0.2189**	0.2674	0.2749
10	trainbr()	**0.0129**	0.4337	0.3371	0.0267

Table 5.19: MSE for *Case*-19 using ten training functions and two AFs

MIX DESIGN for f_{c28}					
		MSE			
AF used in hidden layer→		log-sigmoid		tan-sigmoid	
S. No.	Training Function↓	*hn* = 50	*hn* = 60	*hn* = 50	*hn* = 60
1	trainbfg()	**0.0109**	0.0997	0.0166	0.1311
2	traincgp()	0.0408	0.0256	**0.0036**	0.1191
3	traincgb()	0.0727	**0.0610**	0.0027	0.1308
4	traincgf()	0.0314	0.1319	**0.0109**	0.1227
5	trainlm()	0.0012	0.0016	**0.0001**	0.0013
6	trainrp()	0.1521	0.1468	**0.0195**	0.1001
7	traingda()	**0.0680**	0.1139	0.0178	0.0332
8	trainscg()	**0.0040**	0.1326	0.0113	0.1059
9	trainoss()	0.0541	0.1043	**0.0051**	0.1276
10	trainbr()	**0.0109**	0.0997	0.0166	0.1311

Table 5.20: MSE for *Case*-20 using ten training functions and two AFs

MIX DESIGN for f_{c56}					
		MSE			
AF used in hidden layer→		log-sigmoid		tan-sigmoid	
S. No.	Training Function↓	*hn* = 50	*hn* = 60	*hn* = 50	*hn* = 60
1	trainbfg()	**0.0088**	0.1034	0.0122	0.0747
2	traincgp()	**0.0077**	0.0967	0.0344	0.0982

3	traincgb()	**0.0116**	0.0909	0.0164	0.0944
4	traincgf()	**0.0105**	0.1059	0.0265	0.101
5	trainlm()	**0.0005**	0.0011	0.0016	0.0077
6	trainrp()	0.0756	0.1051	**0.0269**	0.0737
7	traingda()	**0.0052**	0.1109	0.0127	0.1005
8	trainscg()	0.0626	0.0878	**0.0169**	0.1108
9	trainoss()	**0.0123**	0.0994	0.0198	0.1012
10	trainbr()	0.0469	0.0761	**0.0212**	0.1009

Table 5.21: MSE for *Case*-21 using ten training functions and two AFs

MIX DESIGN for f_{c91}					
		MSE			
AF used in hidden layer→		**log-sigmoid**		**tan-sigmoid**	
S. No.	**Training Function↓**	***hn* = 50**	***hn* = 60**	***hn* = 50**	***hn* = 60**
1	trainbfg()	**0.0208**	0.1013	0.0056	0.0116
2	traincgp()	**0.0050**	0.0428	0.0119	0.1181
3	traincgb()	0.0556	**0.0194**	0.0012	0.1009
4	traincgf()	**0.0040**	0.0076	0.0064	0.1215
5	trainlm()	0.0039	0.0127	0.0014	**0.0007**
6	trainrp()	**0.0224**	0.1016	0.0138	0.0512
7	traingda()	0.0031	0.0019	0.0166	**0.0018**
8	trainscg()	**0.0327**	0.1151	0.0209	0.1158
9	trainoss()	**0.0275**	0.0534	0.0119	0.1044
10	trainbr()	**0.0128**	0.1012	0.0034	0.0935

Table 5.22: MSE for *Case*-22 using ten training functions and two AFs

MIX DESIGN for f_{c28}					
		MSE			
AF used in hidden layer→		**log-sigmoid**		**tan-sigmoid**	
S. No.	**Training Function↓**	***hn* = 50**	***hn* = 60**	***hn* = 50**	***hn* = 60**
1	trainbfg()	0.0382	0.1076	0.0108	**0.0070**
2	traincgp()	**0.0110**	0.0385	0.0419	0.1438
3	traincgb()	0.0205	0.1526	**0.0028**	0.2478
4	traincgf()	**0.0080**	0.0776	0.0101	0.396
5	trainlm()	0.0002	0.0002	**0.0001**	0.0009
6	trainrp()	**0.0074**	0.1268	0.0147	0.2503
7	traingda()	**0.0028**	0.0243	0.0432	0.0054
8	trainscg()	0.0139	0.0308	**0.0097**	0.0461
9	trainoss()	**0.0080**	0.0196	0.0125	0.0485

10	trainbr()	0.0305	0.0306	**0.0110**	0.0814

Table 5.23: MSE for *Case*-23 using ten training functions and two AFs

MIX DESIGN for f_{c56}					
		MSE			
AF used in hidden layer→		**log-sigmoid**		**tan-sigmoid**	
S. No.	**Training Function↓**	***hn* = 50**	***hn* = 60**	***hn* = 50**	***hn* = 60**
1	trainbfg()	**0.0034**	0.0141	0.0634	0.0468
2	traincgp()	**0.0049**	0.01	0.0348	0.0947
3	traincgb()	**0.0053**	0.0055	0.0105	0.0403
4	traincgf()	**0.0045**	0.0317	0.0097	0.0231
5	trainlm()	0.0155	0.028	**0.0002**	0.003
6	trainrp()	**0.0252**	0.1411	0.0574	0.051
7	traingda()	0.0054	0.0035	0**.0012**	0.0023
8	trainscg()	**0.0097**	0.0701	0.0127	0.1683
9	trainoss()	0.0234	0.0473	**0.0221**	0.2623
10	trainbr()	**0.0058**	0.08	0.0101	0.066

Table 5.24: MSE for *Case*-24 using ten training functions and two AFs

MIX DESIGN for f_{c91}					
		MSE			
AF used in hidden layer→		**log-sigmoid**		**tan-sigmoid**	
S. No.	**Training Function↓**	***hn* = 50**	***hn* = 60**	***hn* = 50**	***hn* = 60**
1	trainbfg()	0.035	0.0315	**0.0092**	0.0381
2	traincgp()	0.0426	0.0359	**0.007**	0.1217
3	traincgb()	**0.0261**	0.0381	0.0264	0.1382
4	traincgf()	**0.0068**	0.1268	0.0086	0.0335
5	trainlm()	0.0005	0.0003	**0.0001**	0.0028
6	trainrp()	**0.0077**	0.0287	0.0088	0.0183
7	traingda()	--	--	**0.0031**	0.0057
8	trainscg()	**0.0054**	0.033	0.0071	0.0737
9	trainoss()	**0.0139**	0.0331	0.0177	0.0453
10	trainbr()	0.0257	0.0867	**0.0190**	0.1538

Table 5.25: MSE for *Case*-25 using ten training functions and two AFs

MIX DESIGN for f_{c28}					
		MSE			
AF used in hidden layer→		**log-sigmoid**		**tan-sigmoid**	
S. No.	**Training Function↓**	***hn* = 50**	***hn* = 60**	***hn* = 50**	***hn* = 60**
1	trainbfg()	0.0155	0.0310	**0.0085**	0.0095
2	traincgp()	0.0312	0.0357	**0.0057**	0.0097
3	traincgb()	0.0066	0.023	**0.0050**	0.0087
4	traincgf()	0.0143	0.0282	0.0097	**0.0095**
5	trainlm()	0.0043	0.0098	**0.0040**	0.0078
6	trainrp()	0.0132	0.0323	**0.0081**	0.0097
7	traingda()	0.0048	0.0223	0.0025	**0.0012**
8	trainscg()	0.0275	0.0237	**0.0099**	0.0100
9	trainoss()	**0.0087**	0.0197	0.0093	0.0091
10	trainbr()	---	---	---	---

Table 5.26: MSE for *Case*-26 using ten training functions and two AFs

MIX DESIGN for f_{c56}					
		MSE			
AF used in hidden layer→		**log-sigmoid**		**tan-sigmoid**	
S. No.	**Training Function↓**	***hn* = 50**	***hn* = 60**	***hn* = 50**	***hn* = 60**
1	trainbfg()	0.0117	0.0172	0.0086	**0.0083**
2	traincgp()	--	--	--	--
3	traincgb()	--	--	--	--
4	traincgf()	--	--	--	--
5	trainlm()	**0.0034**	0.0196	0.0060	0.0037
6	trainrp()	0.0261	0.0230	0.0099	**0.0099**
7	traingda()	0.0028	0.0056	**0.0014**	0.0031
8	trainscg()	0.0288	0.0276	**0.0100**	**0.0100**
9	trainoss()	0.0104	0.0305	**0.0093**	0.0094
10	trainbr()	--	--	--	--

Table 5.27: MSE for *Case*-27 using ten training functions and two AFs

MIX DESIGN for f_{c91}					
		MSE			
AF used in hidden layer→		**log-sigmoid**		**tan-sigmoid**	
S. No.	**Training Function↓**	***hn* = 50**	***hn* = 60**	***hn* = 50**	***hn* = 60**
1	trainbfg()	0.0089	0.0368	0.0169	**0.0088**
2	traincgp()	--	--	--	--

3	traincgb()	--	--	--	--
4	traincgf()	--	--	--	--
5	trainlm()	0.0071	0.0920	**0.0005**	0.0079
6	trainrp()	0.0085	0.0435	**0.0085**	0.0098
7	traingda()	0.0028	**0.0012**	0.0024	0.06
8	trainscg()	**0.0094**	0.0171	0.0161	0.0100
9	trainoss()	0.0136	0.0666	**0.0065**	0.0100
10	trainbr()	--	--	--	--

Table 5.28: MSE for *Case*-28 using ten training functions and two AFs

MIX DESIGN for f_{c28}					
		MSE			
AF used in hidden layer→		**log-sigmoid**		**tan-sigmoid**	
S. No.	**Training Function↓**	***hn* = 50**	***hn* = 60**	***hn* = 50**	***hn* = 60**
1	trainbfg()	0.0076	0.0277	**0.0066**	0.0098
2	traincgp()	--	--	--	--
3	traincgb()	--	--	--	--
4	traincgf()	--	--	--	--
5	trainlm()	0.0096	0.0110	**0.0074**	0.0081
6	trainrp()	0.0272	0.0346	**0.0099**	0.0100
7	traingda()	0.0037	0.0055	**0.0028**	0.0031
8	trainscg()	0.0312	**0.0081**	0.0100	0.0100
9	trainoss()	0.0114	0.0311	0.0097	**0.0073**
10	trainbr()	--	--	--	--

Table 5.29: MSE for *Case*-29 using ten training functions and two AFs

MIX DESIGN for f_{c56}					
		MSE			
AF used in hidden layer→		**log-sigmoid**		**tan-sigmoid**	
S. No.	**Training Function↓**	***hn* = 50**	***hn* = 60**	***hn* = 50**	***hn* = 60**
1	trainbfg()	0.0291	0.1062	**0.0098**	0.2461
2	traincgp()	0.1090	0.5246	**0.0076**	0.5428
3	traincgb()	0.01021	0.2651	**0.0093**	0.4799
4	traincgf()	0.0958	0.3047	**0.0085**	0.1958
5	trainlm()	0.0025	0.3683	**3.7191e-004**	0.0404
6	trainrp()	--	--	**0.0095**	0.6980
7	traingda()	--	--	--	--
8	trainscg()	--	--	**0.0099**	0.6772
9	trainoss()	0.1090	0.0648	**0.0094**	0.2536

10	trainbr()	--	--	--	--

Table 5.30: MSE for *Case*-30 using ten training functions and two AFs

MIX DESIGN for f_{c91}					
		MSE			
AF used in hidden layer→		**log-sigmoid**		**tan-sigmoid**	
S. No.	**Training Function↓**	***hn* = 50**	***hn* = 60**	***hn* = 50**	***hn* = 60**
1	trainbfg()	0.0243	0.0276	**0.0136**	0.0291
2	traincgp()	--	--	--	--
3	traincgb()	--	--	--	--
4	traincgf()	--	--	--	--
5	trainlm()	0.0096	0.0099	**0.0063**	0.0198
6	trainrp()	0.0111	0.015	**0.0073**	0.0225
7	traingda()	--	--	--	--
8	trainscg()	--	--	--	--
9	trainoss()	0.0098	0.0091	**0.0052**	0.0167
10	trainbr()	--	--	--	--

Table 5.31 MSE for *Case*-31 using ten training functions and two AFs

MIX DESIGN for f_{c28}					
		MSE			
AF used in hidden layer→		**log-sigmoid**		**tan-sigmoid**	
S. No.	**Training Function↓**	***hn* = 50**	***hn* = 60**	***hn* = 50**	***hn* = 60**
1	trainbfg()	0.1654	0.1342	**0.1147**	0.1387
2	traincgp()	--	---	--	--
3	traincgb()	--	---	---	---
4	traincgf()	--	---	--	--
5	trainlm()	0.0082	0.0095	**0.0070**	0.1468
6	trainrp()	**0.0100**	0.0100	0.0610	0.1821
7	traingda()	--	--	--	--
8	trainscg()	--	--	--	--
9	trainoss()	--	--	--	--
10	trainbr()	--	--	--	--

Table 5.32: MSE for *Case*-32 using ten training functions and two AFs

MIX DESIGN for f_{c56}					
		MSE			
AF used in hidden layer→		**log-sigmoid**		**tan-sigmoid**	
S. No.	**Training Function↓**	***hn* = 50**	***hn* = 60**	***hn* = 50**	***hn* = 60**
1	trainbfg()	0.2341	0.2231	0.2103	**0.2038**
2	traincgp()	--	--	--	--
3	traincgb()	--	--	--	--
4	traincgf()	--	--	--	--
5	trainlm()	0.0093	**0.0067**	0.0073	0.0520
6	trainrp()	--	--	--	--
7	traingda()	--	--	--	--
8	trainscg()	--	--	--	--
9	trainoss()	**0.0078**	0.0100	0.012	0.0276
10	trainbr()	--	--	--	--

Table 5.33: MSE for *Case*-33 using ten training functions and two AFs

MIX DESIGN for f_{c91}					
		MSE			
AF used in hidden layer→		**log-sigmoid**		**tan-sigmoid**	
S. No.	**Training Function↓**	***hn* = 50**	***hn* = 60**	***hn* = 50**	***hn* = 60**
1	trainbfg()	**0.2841**	0.3014	0.2984	0.3192
2	traincgp()	--	--	--	--
3	traincgb()	--	--	--	--
4	traincgf()	--	--	--	--
5	trainlm()	0.0039	**0.0019**	0.0063	0.0055
6	trainrp()	0.0100	**0.0001**	0.0025	0.0176
7	traingda()	--	--	--	--
8	trainscg()	--	--	--	--
9	trainoss()	0.0099	0.0100	**0.0092**	0.0448
10	trainbr()	--	--	--	--

Table 5.34: MSE for *Case*-34 using ten training functions and two AFs

MIX DESIGN for f_{c28}					
		MSE			
AF used in hidden layer→		**log-sigmoid**		**tan-sigmoid**	
S. No.	**Training Function↓**	***hn* = 50**	***hn* = 60**	***hn* = 50**	***hn* = 60**
1	trainbfg()	**0.1782**	0.1983	0.2784	0.2834
2	traincgp()	--	--	--	--

3	traincgb()	--	--	--	--
4	traincgf()	--	--	--	--
5	trainlm()	0.0065	0.0056	**0.0052**	0.0667
6	trainrp()	**0.0100**	**0.0100**	0.0642	0.0707
7	traingda()	0.0030	0.0031	0.0309	0.0280
8	trainscg()	--	--	--	--
9	trainoss()	0.0095	0.0099	0.0332	0.0492
10	trainbr()	--	--	--	--

Table 5.35: MSE for *Case*-35 using ten training functions and two AFs

MIX DESIGN for f_{c56}					
		MSE			
AF used in hidden layer→		**log-sigmoid**		**tan-sigmoid**	
S. No.	**Training Function↓**	***hn* = 50**	***hn* = 60**	***hn* = 50**	***hn* = 60**
1	trainbfg()	0.1985	**0.1749**	0.1830	0.1915
2	traincgp()	--	--	--	--
3	traincgb()	--	--	--	--
4	traincgf()	--	--	--	--
5	trainlm()	0.0096	**0.0007**	0.0035	0.0703
6	trainrp()	0.0100	**0.0089**	0.0351	0.1287
7	traingda()	--	--	--	--
8	trainscg()	--	--	--	--
9	trainoss()	**0.0099**	0.0101	0.0583	0.0502
10	trainbr()	--	--	--	--

Table 5.36: MSE for *Case*-36 using ten training functions and two AFs

MIX DESIGN for f_{c91}					
		MSE			
AF used in hidden layer→		**log-sigmoid**		**tan-sigmoid**	
S. No.	**Training Function↓**	***hn* = 50**	***hn* = 60**	***hn* = 50**	***hn* = 60**
1	trainbfg()	0.1654	0.1453	0.1847	**0.1238**
2	traincgp()	--	--	--	--
3	traincgb()	--	--	--	--
4	traincgf()	--	--	--	--
5	trainlm()	0.0041	0.0058	**0.0003**	0.0345
6	trainrp()	**0.0100**	0.0100	0.0415	0.0108
7	traingda()	--	--	--	--
8	trainscg()	--	--	--	--
9	trainoss()	**0.0094**	0.0100	0.0285	0.0332

10	trainbr()	--	--	--	--

It has been seen during experimentation that smaller value of *lr* increases the convergence time and the model takes more time in learning. As such, in some cases, the maximum number of epochs was exhausted, and the model could not converge. After a critical analysis of these tables, we have finalized the values of these parameters and the same are depicted in Table 5.37. We have also excluded two training functions, namely, trainbr() and traingda() from further experimentation as the models could not generally coverage while using these functions. As such, we have considered remaining eight training functions in the further experiments.

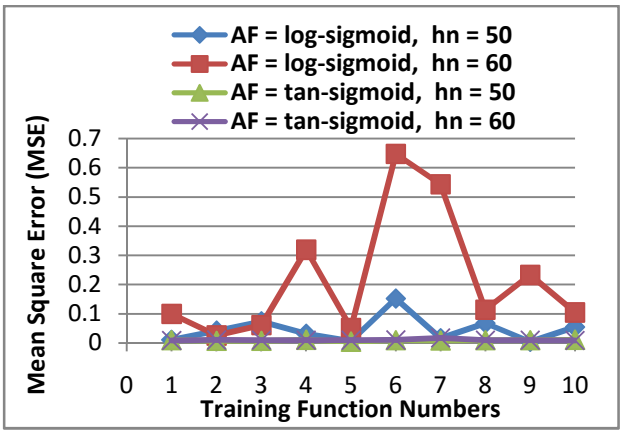

Fig. 5.3: MSE achieved for different training functions for *Case*-1

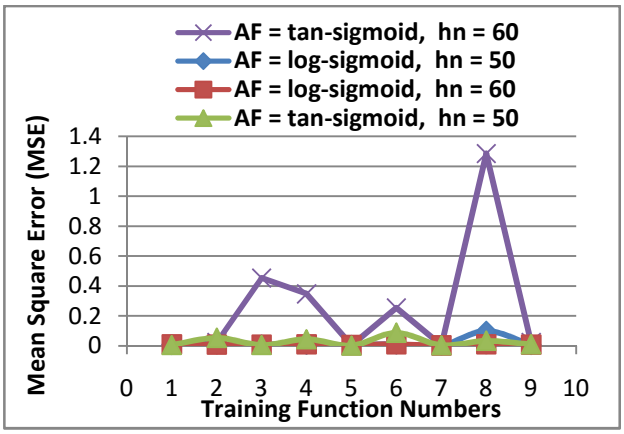

Fig. 5.4: MSE achieved for different training functions for *Case*-2

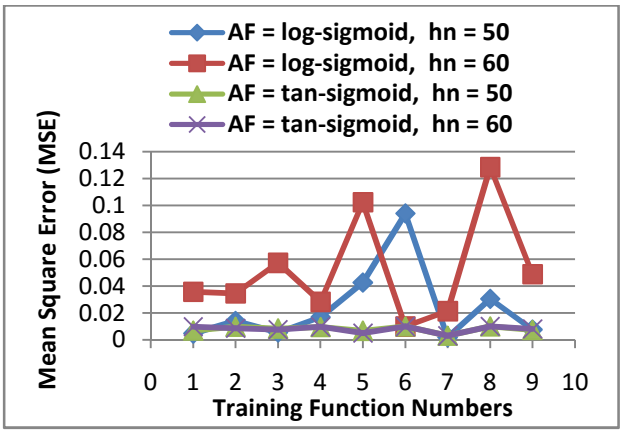

Fig. 5.5: MSE achieved for different training functions for *Case*-3

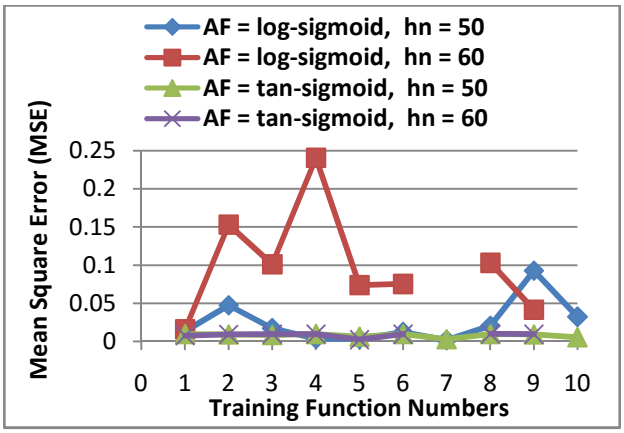

Fig. 5.6: MSE achieved for different training functions for *Case*-4

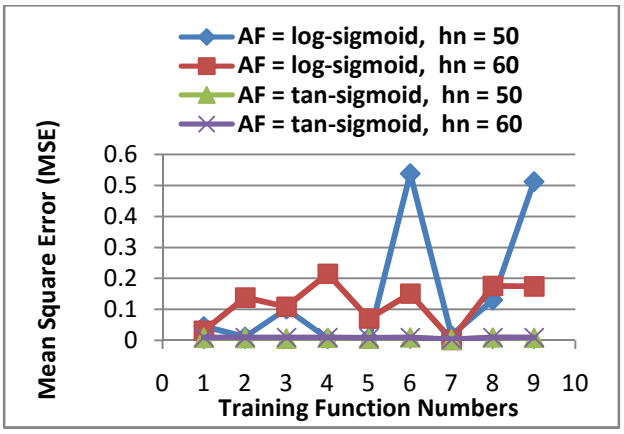

Fig. 5.7: MSE achieved for different training functions for *Case*-5

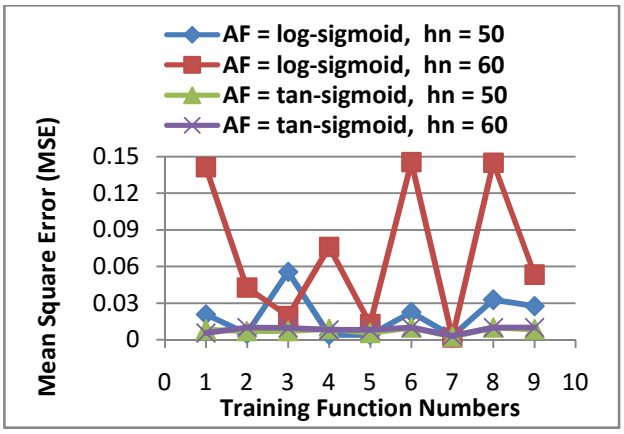

Fig. 5.8: MSE achieved for different training functions for *Case*-6

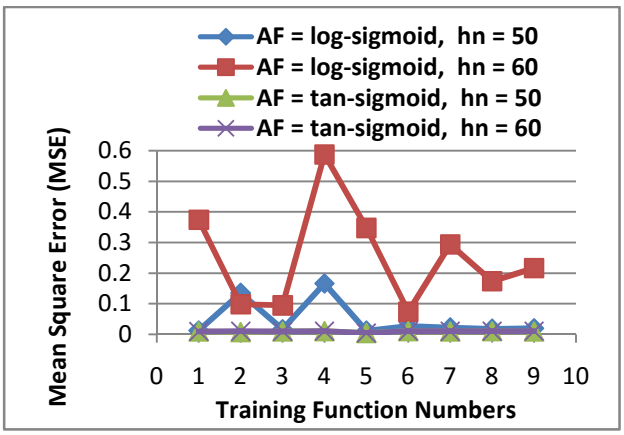

Fig. 5.9: MSE achieved for different training functions for *Case*-7

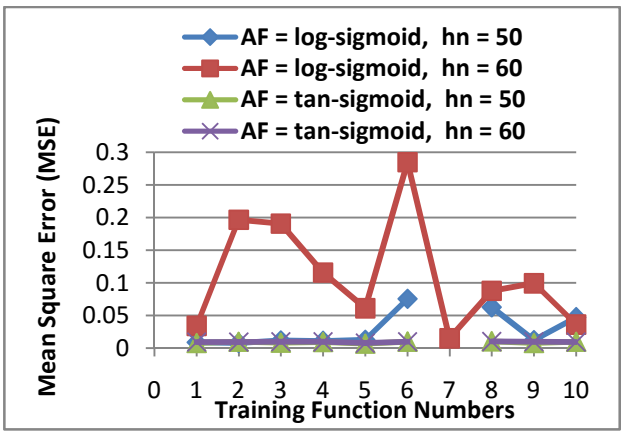

Fig. 5.10: MSE achieved for different training functions for *Case*-8

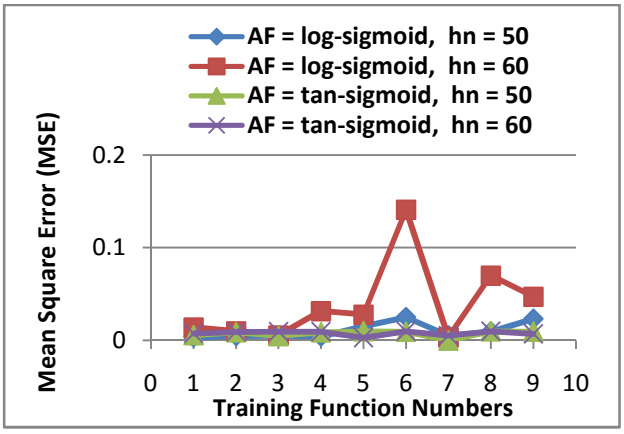

Fig. 5.11: MSE achieved for different training functions for *Case*-9

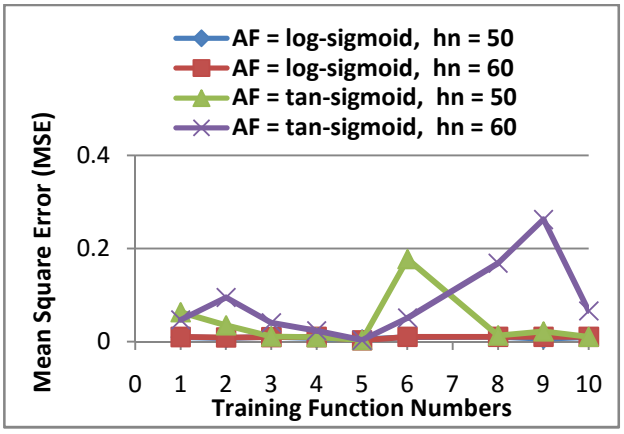

Fig. 5.12: MSE achieved for different training functions for *Case*-10

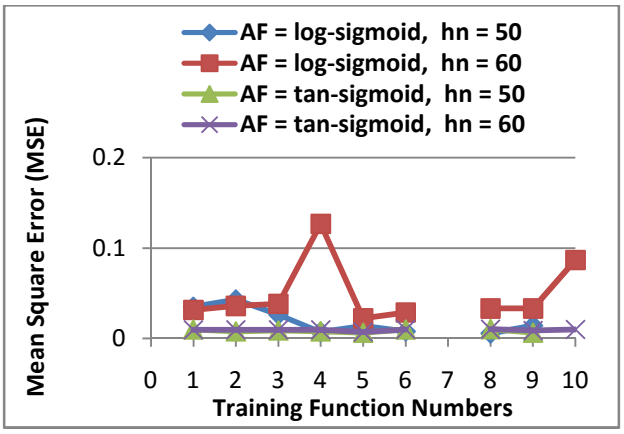

Fig. 5.13: MSE achieved for different training functions for *Case*-11

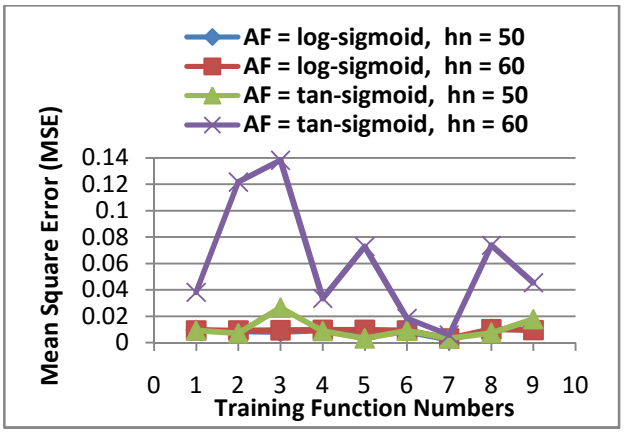

Fig. 5.14: MSE achieved for different training functions for *Case*-12

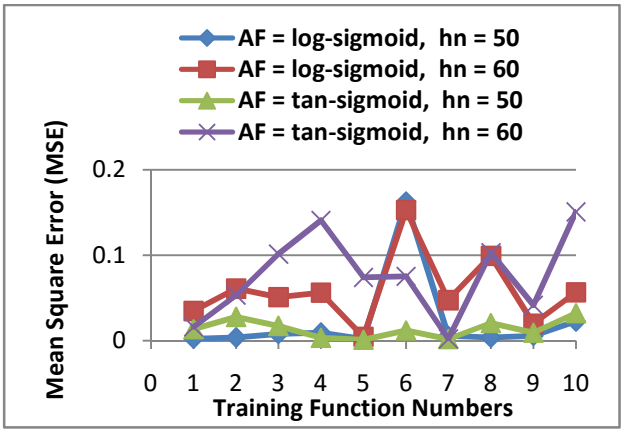

Fig. 5.15: MSE achieved for different training functions for *Case*-13

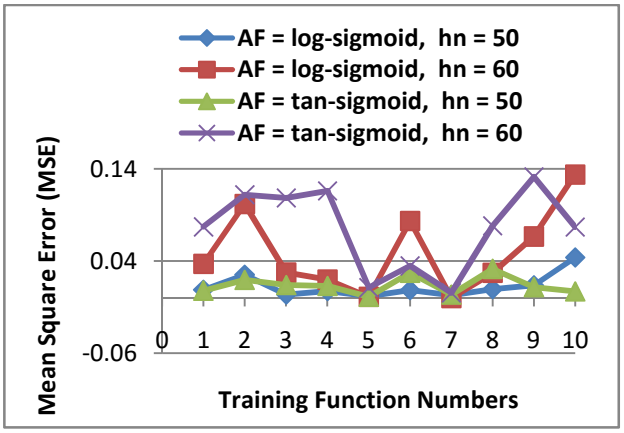

Fig. 5.16: MSE achieved for different training functions for *Case*-14

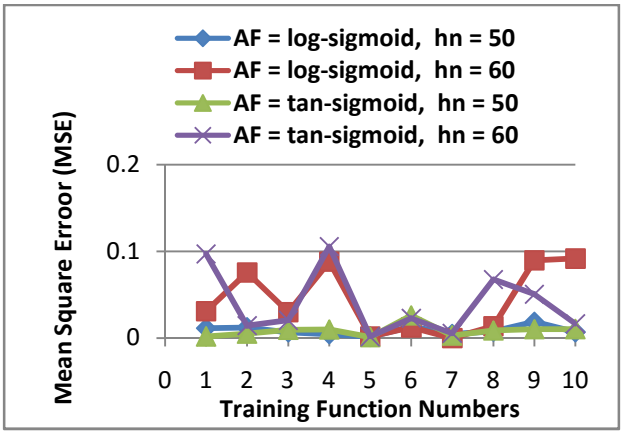

Fig. 5.17: MSE achieved for different training functions for *Case*-15

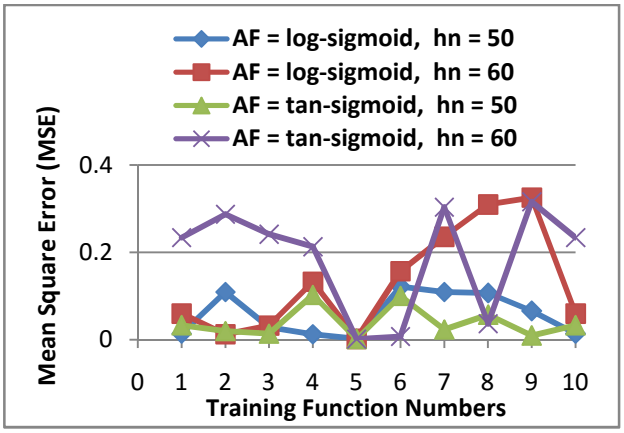

Fig. 5.18: MSE achieved for different training functions for *Case*-16

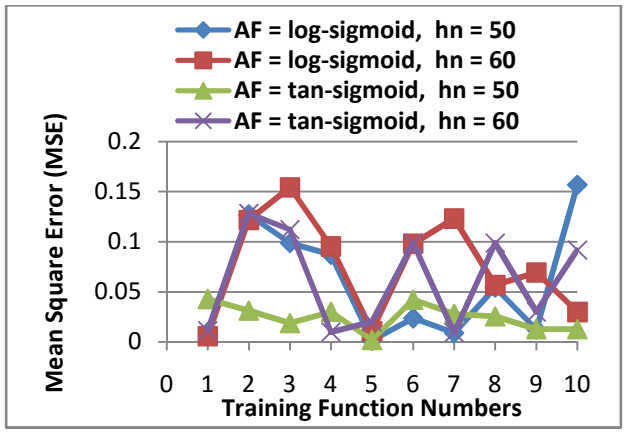

Fig. 5.19: MSE achieved for different training functions for *Case*-17

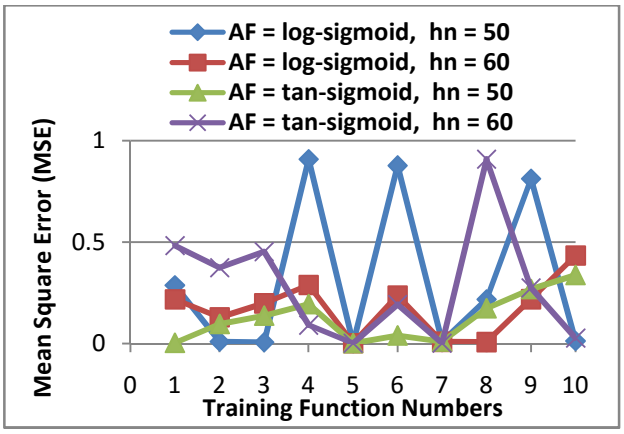

Fig. 5.20: MSE achieved for different training functions for *Case*-18

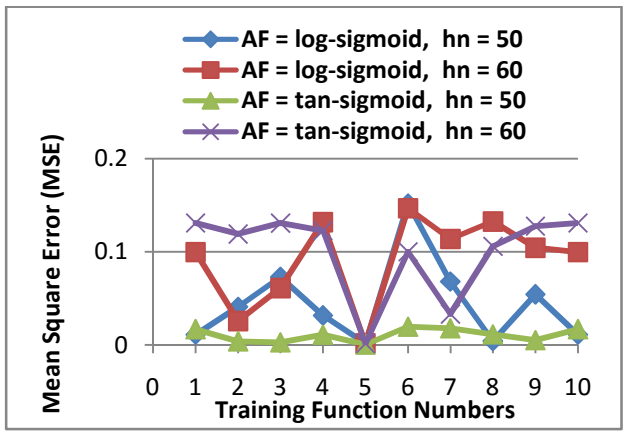

Fig. 5.21: MSE achieved for different training functions for *Case*-19

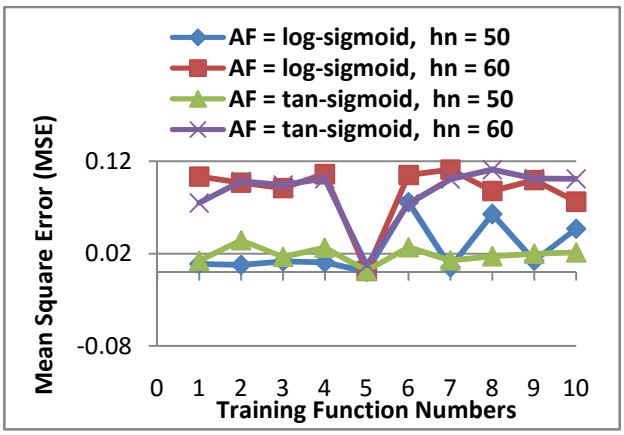

Fig. 5.22: MSE achieved for different training functions for *Case*-20

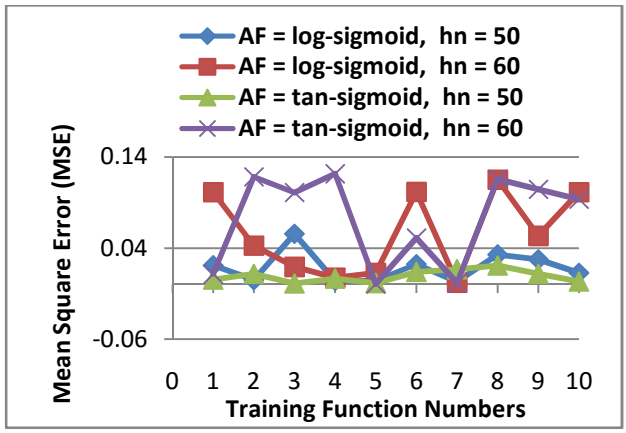

Fig. 5.23: MSE achieved for different training functions for *Case*-21

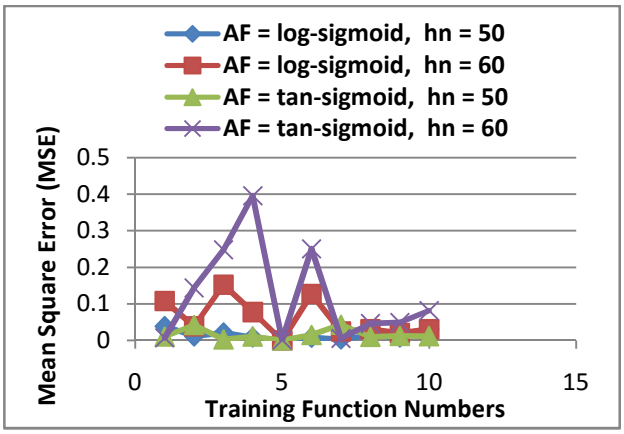

Fig. 5.24: MSE achieved for different training functions for *Case*-22

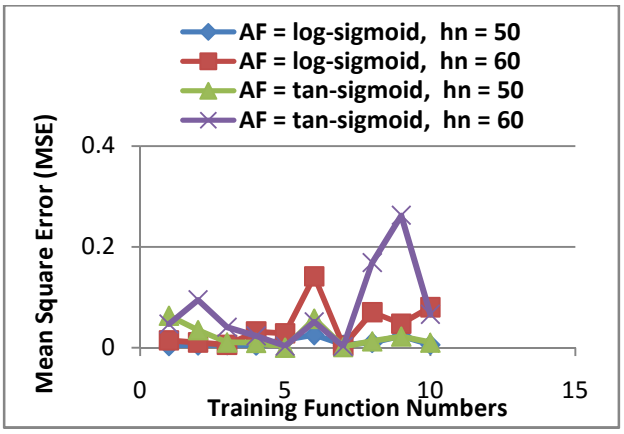

Fig. 5.25: MSE achieved for different training functions for *Case*-23

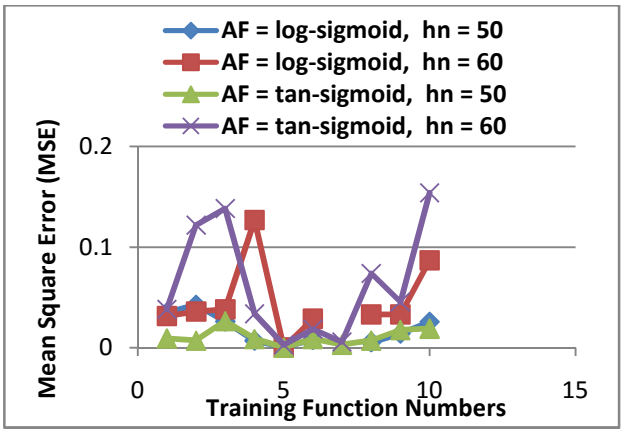

Fig. 5.26: MSE achieved for different training functions for *Case*-24

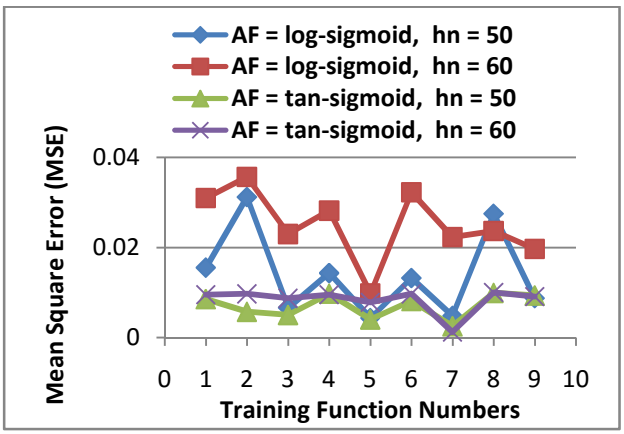

Fig. 5.27: MSE achieved for different training functions for *Case*-25

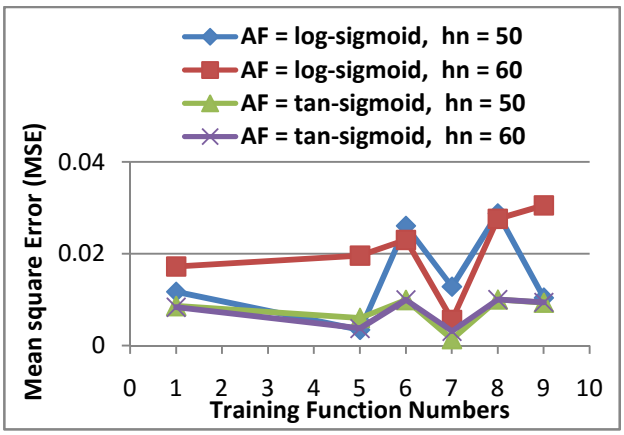

Fig. 5.28: MSE achieved for different training functions for *Case*-26

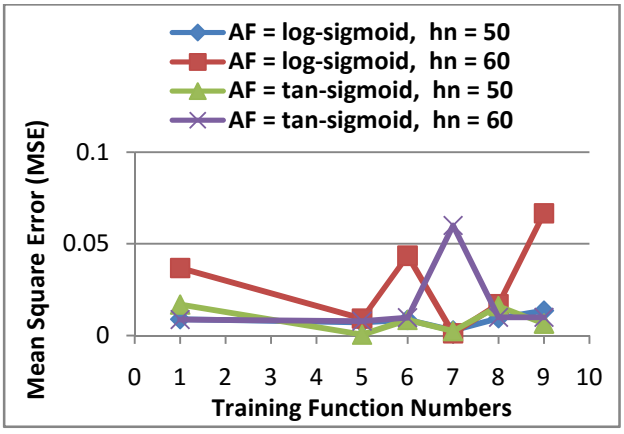

Fig. 5.29: MSE achieved for different training functions for *Case*-27

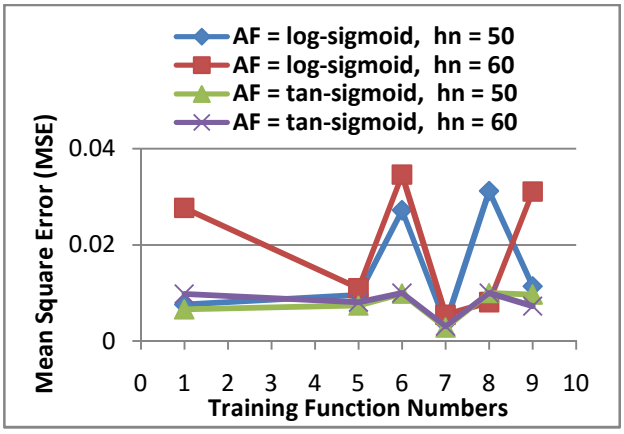

Fig. 5.30: MSE achieved for different training functions for *Case*-28

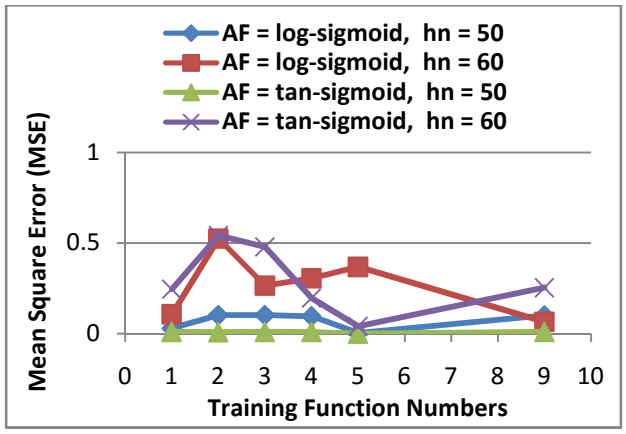

Fig. 5.31: MSE achieved for different training functions for *Case*-29

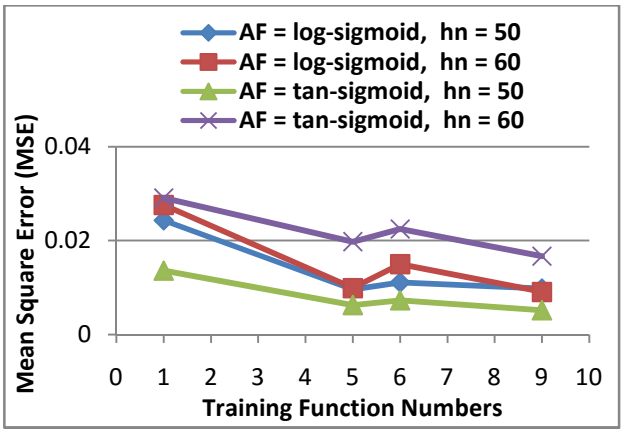

Fig. 5.32: MSE achieved for different training functions for *Case*-30

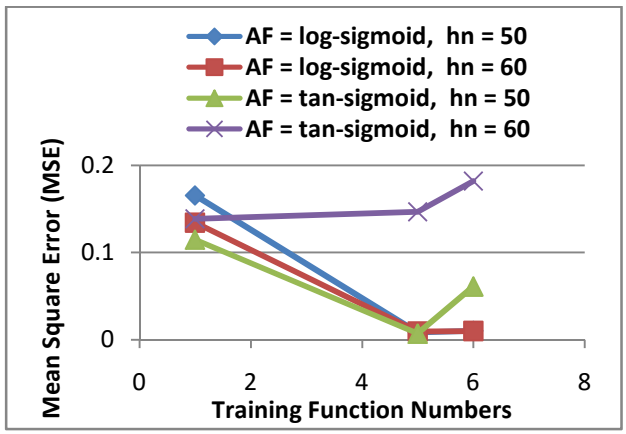

Fig. 5.33: MSE achieved for different training functions for *Case*-31

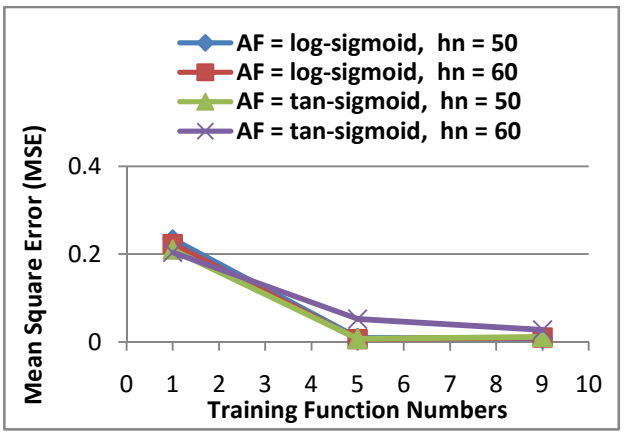

Fig. 5.34: MSE achieved for different training functions for *Case*-32

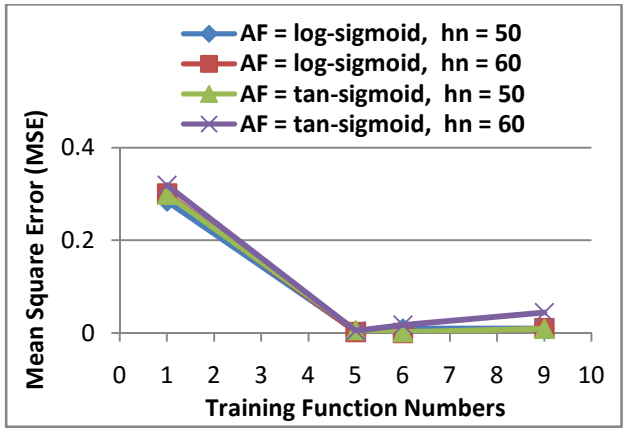

Fig. 5.35: MSE achieved for different training functions for *Case*-33

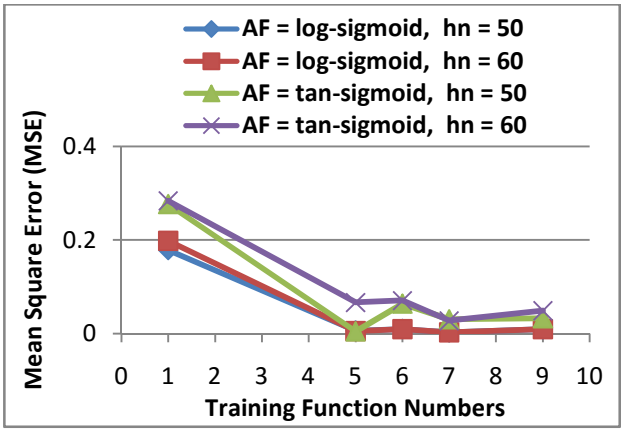

Fig. 5.36: MSE achieved for different training functions for *Case*-34

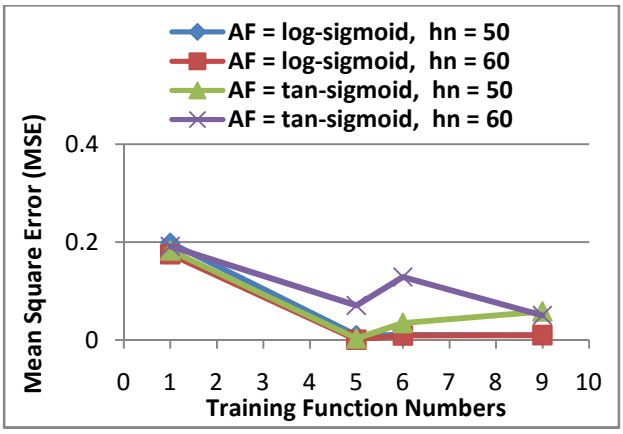

Fig. 5.37: MSE achieved for different training functions for *Case*-35

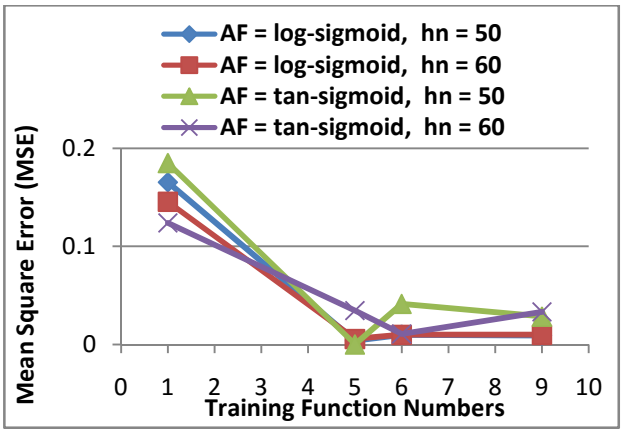

Fig. 5.38: MSE achieved for different training functions for *Case*-36

5.3 Development of neural network model

Table 5.37 depicts the values of parameters used in neural networks for the prediction of compressive strength of concrete. Experiments have further been carried out using eight different training functions and two AFs. For these experiments, the dataset is divided into two disjoint subsets, *i.e.,* with and without the substitution of cement with FA. Dataset - 1 consists of 49 tuples without FA and Dataset - 2 consists of 27 tuples with FA. The values of *lr*, *goal* and *hn* for hidden layer has been taken as 0.01, 0.000001 and 50, respectively. MATLAB routine post-regression (postreg()) has been used to measure the network performance, which implements a regression analysis between network response and corresponding targets.

Table 5.37: Parameters used to develop neural network architecture

Parameters	Values	Description
Input	*w/cm*	Water-cement ratio
	fa/cm	Fine aggregate-cement ratio
	ca/cm	Coarse aggregate-cement ratio
Output	f_{c28}	Compressive strength of 28 curing days
	f_{c56}	Compressive strength of 56 curing days
	f_{c91}	Compressive strength of 91 curing days
Dataset - 1	49 tuples	Without FA (cement is not replaced by FA)
Dataset - 2	27 tuples	With FA (15% of the cement is replaced by FA)
AF-1	tan-sigmoid	$\text{tansig}(x) = \frac{2}{1+e^{-2x}} - 1$

AF-2	log-sigmoid	$\text{logsig}(x) = \frac{1}{1+e^{-x}}$
AF-3	linear	$\text{purelin}(x) = x$
Performance function	MSE	Performance function (Mean Square Error)
Net.trainparam.lr	0.01	Learning rate
Net.trainfcn	trainbfg()	Quasi-Newton algorithm with Broyden, Fletcher, Goldfarb, and Shanno (BFGS) update (BFG)
	traincgp()	Polak-Ribiere conjugate gradient algorithm (CGP)
	traincgb()	Powell-Beale conjugate gradient algorithm (CGB)
	traincgf()	Fletcher-reeves conjugate gradient algorithm (CGF)
	trainlm()	Levenberg–Marquardt (LM)
	trainrp()	Resilient backpropagation (RP)
	trainscg()	Scaled conjugate gradient (SCG)
	trainoss()	One step Secant backpropagation (OSS)
Net.trainparam.epochs	10000	Maximum number of epochs fixed in training
Net.trainparam.goal	0.000001	Performance goal
Net.trainparam.show	15	Epochs between displays
Number of hidden layer neurons	50	Number of neurons in the hidden layer
Number of output layer neurons	01	Number of neurons in the output layer

5.3.1 Prediction of compressive strength of concrete without FA and with 15% FA

Twelve experiments (Exp - (I - XII)) have been performed for the prediction of compressive strength of concrete with FA and without FA using neural network models. The metadata for these experiments is given below:

Exp - I: Dataset - 1, AF is log-sigmoid; Output if f_{c28}

Exp - II: Dataset - 1, AF is log-sigmoid; Output if f_{c56}

Exp - III: Dataset - 1, AF is log-sigmoid; Output if f_{c91}

Exp - IV: Dataset - 1, AF is tan-sigmoid; Output if f_{c28}

Exp - V: Dataset - 1, AF is tan-sigmoid; Output if f_{c56}

Exp - VI: Dataset - 1, AF is tan-sigmoid; Output if f_{c91}

Exp - VII: Dataset - 2, AF is log-sigmoid; Output if f_{c28}

Exp - VIII: Dataset - 2, AF is log-sigmoid; Output if f_{c56}

Exp - IX: Dataset - 2, AF is log-sigmoid; Output if f_{c91}

Exp - X: Dataset - 2, AF is tan-sigmoid; Output if f_{c28}

Exp - XI: Dataset - 2, AF is tan-sigmoid; Output if f_{c56}

Exp - XII: Dataset - 2, AF is tan-sigmoid; Output if f_{c91}

5.3.2 Results and discussions

The results of Exp - I to Exp - VI are presented in Table 5.38 and results of Exp - VII to Exp - XII in Table 5.39. The prediction models are graphically depicted in Figs. 5.39 - 5.134. Exp - I to Exp - III have been performed using the log-sigmoid as AF for f_{c28}, f_{c56} and $f_{c91,}$ respectively, with Dataset - 1. The post-regression fits for Exp - I where f_{c28} is being predicted are presented in Figs. 5.39 - 5.46. It can be seen from these figures that correlation coefficient (R) is more than 0.909 for all training functions except for traincgp(), traincgb() and traincgf(); in which R is 0.734, 0.874 and 0.853, respectively. In Exp - I, the best value of R (= 0.958) is in case of trainlm(). For Exp - II, the post-regression fits are presented in Figs. 5.47 - 5.54. In this experiment also, the best training function is trainlm() and its respective value of R is 0.935. Other training functions yields a value of R >= 0.879 except for traincgp(). Exp - III is for prediction of f_{c91}. Here, the lowest value of R is 0.771 for traincgf() and the highest value of R (= 0.922) is for trainlm. The post-regression fits for f_{c91} are presented in Figs. 5.55 - 5.62.

As mentioned above, Exp - IV to Exp - VI have been conducted using tan-sigmoid as AF and with Dataset - 1. Exp - IV is for the prediction of f_{c28} and the post-regression fits for this experiment are presented in Figs. 5.63 - 5.70. The best value of R (= 0.969) here is again in case of training function trainlm(). However, other training functions have also produced a correlation more than 0.910, except the function traincgp() where the value of R is 0.867. The post-regression fits for f_{c56} are presented in Figs. 5.71 - 5.78. In this experiment, all the training functions yield a value of R >= 0.905 except for the function traincgp(). The function trainlm() yields a perfect correlation of 1.000. In Exp - VI where f_{c91} is being predicted, the best fit is for trainlm() where the value of R is 0.957. Other training functions attain a value of R >= 0.901

except the function traincgf() where the value of R is 0.883. The post-regression fits for f_{c91} are depicted in Figs. 5.79 - 5.86.

The log-sigmoid and tan-sigmoid functions have also been used with Dataset - 2 in Exp - (VII - IX) and Exp - (X - XII), respectively. In all the experiments with Dataset - 2, the value of R comes out to be more than 0.912. Exp - VII is for the prediction of f_{c28}. These results are graphically depicted in Figs. 5.87 - 5. 94. Here the trend repeats and the best training function is again trainlm() with R = 0.980. For Exp - VIII in which f_{c56} being predicted, the post-regression fits are presented in Figs. 5.95 - 5.102. The optimal value of R (= 0.954) is for trainlm() and the value of R for all other training functions is more than 0.937. Exp - IX is conducted for the preiction of f_{c91}. The post-regression fits are presented in Figs. 5.103 - 5.110. The highsest value of R here is 1.000 for trainlm(). The post-regression fits for the prediction of f_{c28} (Exp - X) are depicted in Figs. 5.111 - 5.118. All the training functions here attain a value of R >= 0.917 and the highest value of R (= 0.989) is achieved for trainlm(). Exp - XI has been performed for the prediction of f_{c56}. Here, the value of R (= 0.983) is highest for again trainlm(). All other training functions are also predicting well with R >= 0.937. Graphical presentation of post-regression fits for this are depicted in Figs. 5.119 - 5.126. Exp - XII has been performed for the prediction of f_{c91}. The post-regression fits for this experiment are presented in Figs. 5.127 - 5.134. The value of R is more than 0.922 for all the training functions and the highest value of R (= 0.996) is again achieved for trainlm() for this experiment also.

Table 5.38: Prediction performance of various neural network models trained using eight different training functions for Dataset - 1

AF = log-sigmoid				
Experiment	**Training function**	***R* (Correlation)**	**Epochs required in training**	**Best linear fit given by post-regression (A = predicted strength, T = target strength)**
Exp - I	trainbfg()	0.909	118	A = (0.896)T + (0.0112)
	traincgp()	0.734	252	A = (0.924)T + (0.00834)
	traincgb()	0.874	237	A = (0.926)T + (0.00788)
	traincgf()	0.853	506	A = (0.882)T + (0.0127)
	trainlm()	0.958	03	A = (0.903)T + (0.0106)
	trainrp()	0.909	4607	A = (0.922)T + (0.00839)
	trainscg()	0.911	493	A = (0.933)T + (0.00717)

	trainoss()	0.909	1208	A = (0.905)T + (0.0102)
Exp - II	trainbfg()	0.897	112	A = (0.872)T + (0.0154)
	traincgp()	0.734	252	A = (0.878)T + ().0147)
	traincgb()	0.898	279	A = (0.916)T + (0.0101)
	traincgf()	0.893	392	A = (0.87)T + (0.0157)
	trainlm()	0.935	04	A = (0.858)T + (0.0163)
	trainrp()	0.894	4294	A = (0.887)T + (0.0136)
	trainscg()	0.897	483	A = (0.911)T + (0.0108)
	trainoss()	0.897	1037	A = (0.914)T + (0.0103)
Exp - III	trainbfg()	0.896	79	A = (0.849)T + (0.0191)
	traincgp()	0.854	408	A = (0.861)T + (0.0178)
	traincgb()	0.894	252	A = (0.865)T + (0.0171)
	traincgf()	0.771	257	A = (0.883)T + (0.015)
	trainlm()	0.922	08	A = (0.832)T + (0.02)
	trainrp()	0.897	3615	A = (0.903)T + (0.0123)
	trainscg()	0.898	318	A = (0.894)T + (0.0134)
	trainoss()	0.894	1062	A = (0.864)T + (0.0172)
AF = tan-sigmoid				
Exp - IV	trainbfg()	0.912	94	A = (0.907)T + (0.00984)
	traincgp()	0.867	252	A = (0.936)T + (0.00685)
	traincgb()	0.916	303	A = (0.976)T + (0.00248)
	traincgf()	0.863	252	A = (0.904)T + (0.0106)
	trainlm()	0.969	07	A = (0.939)T + (0.00663)
	trainrp()	0.910	8457	A = (0.927)T + (0.00782)
	trainscg()	0.912	478	A = (0.939)T + (0.00658)
	trainoss()	0.916	612	A = (0.926)T + (0.00759)
Exp - V	trainbfg()	0.915	79	A = (1.02)T + (-0.00256)
	traincgp()	0.801	302	A = (0.952)T + (0.00589)
	traincgb()	0.913	227	A = (1.02)T + (-0.00265)
	traincgf()	0.906	551	A = (0.975)T + (0.0031)
	trainlm()	1	05	A = (1)T + (-6.17e-005)
	trainrp()	0.908	21325	A = (0.991)T + (0.00105)
	trainscg()	0.905	441	A = (0.968)T + (0.00376)
	trainoss()	0.916	961	A = (0.989)T + (0.00138)
Exp - VI	trainbfg()	0.911	109	A = (1)T + (-0.000634)
	traincgp()	0.905	413	A = (0.971)T + (0.00353)
	traincgb()	0.911	295	A = (1)T + (-0.000207)
	traincgf()	0.883	604	A = (1.01)T + (-0.00099)
	trainlm()	0.957	04	A = (1.04)T + (-0.0044)

	trainrp()	0.908	4698	A = (0.991)T + (0.0011)
	trainscg()	0.908	464	A = (0.989)T + (0.00131)
	trainoss()	0.901	1556	A = (0.913)T + (0.0104)

Table 5.39: Prediction performance of various neural network models trained using eight different training functions for Dataset - 2

AF = log-sigmoid				
Experiment	**Training functions**	***R* (Correlation)**	**Epochs required in training**	**Best linear fit given by post-regression (A = predicted strength, T = target strength)**
Exp - VII	trainbfg()	0.936	41	A = (1.04)T + (-0.00358)
	traincgp()	0.920	164	A = (0.977)T + (0.00205)
	traincgb()	0.928	177	A = (1.04)T + (-0.00366)
	traincgf()	0.925	212	A = (1.02)T + (-0.00238)
	trainlm()	0.980	04	A = (1.11)T + (-0.00815)
	trainrp()	0.921	896	A = (1)T + (-0.00037)
	trainscg()	0.912	130	A = (0.926)T + (0.00686)
	trainoss()	0.915	443	A = (0.925)T + (0.00719)
Exp - VIII	trainbfg()	0.940	73	A = (0.963)T + (0.004)
	traincgp()	0.941	240	A = (1.01)T + (-0.00117)
	traincgb()	0.940	146	A = (0.995)T + (0.000551)
	traincgf()	0.941	246	A = (1)T + (-7.79e-005)
	trainlm()	0.954	02	A = (0.985)T + (0.000663)
	trainrp()	0.939	2384	A = (0.997)T + (0.000285)
	trainscg()	0.937	243	A = (0.976)T + (0.00235)
	trainoss()	0.939	497	A = (0.993)T + (0.00055)
Exp - IX	trainbfg()	0.925	66	A = (1)T + (-8.58e-005)
	traincgp()	0.925	228	A = (0.995)T + (0.000631)
	traincgb()	0.927	130	A = (1)T + (-0.000284)
	traincgf()	0.924	338	A = (0.99)T + (0.000891)
	trainlm()	1	03	A = (1)T + (3.73e-005)
	trainrp()	0.922	3713	A = (0.979)T + (0.00244)
	trainscg()	0.919	147	A = (0.952)T + (0.00566)
	trainoss()	0.931	500	A = (1.03)T + (-0.00406)
AF = tan-sigmoid				
Exp - X	trainbfg()	0.920	73	A = (0.988)T + (0.00095)
	traincgp()	0.917	186	A = (0.962)T + (0.00356)
	traincgb()	0.927	116	A = (1.01)T + (-0.000917)

	traincgf()	0.925	135	A = (0.959)T + (0.00427)
	trainlm()	0.989	03	A = (1)T + (-0.000852)
	trainrp()	0.923	821	A = (1.02)T + (-0.00156)
	trainscg()	0.923	120	A = (1)T + (-0.000125)
	trainoss()	0.923	151	A = (0.939)T + (0.00553)
Exp - XI	trainbfg()	0.953	52	A = (0.965)T + (0.00433)
	traincgp()	0.938	202	A = (0.989)T + (0.00108)
	traincgb()	0.941	231	A = (1.01)T + (-0.000687)
	traincgf()	0.941	245	A = (1.01)T + (-0.00088)
	trainlm()	0.983	03	A = (1.03)T + (-0.0048)
	trainrp()	0.939	2548	A = (0.995)T + (0.00052)
	trainscg()	0.937	202	A = (0.979)T + (0.00227)
	trainoss()	0.949	223	A = (1.02)T + (-0.00235)
Exp - XII	trainbfg()	0.930	38	A = (0.989)T + (0.00115)
	traincgp()	0.929	226	A = (1.02)T + (-0.00218)
	traincgb()	0.926	145	A = (0.995)T + (0.000598)
	traincgf()	0.922	273	A = (0.974)T + (0.00294)
	trainlm()	0.996	03	A = (1.05)T + (-0.00568)
	trainrp()	0.922	1347	A = (0.973)T + (0.00315)
	trainscg()	0.924	150	A = (0.969)T + (0.00322)
	trainoss()	0.924	204	A = (0.993)T + (0.000851)

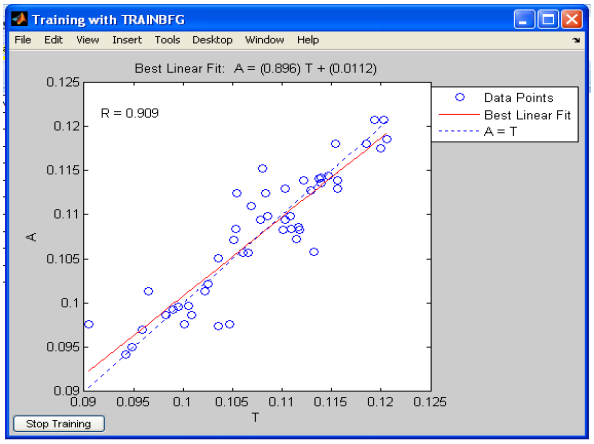

Fig. 5.39: Predicted strength vs. target strength for Exp - I using trainbfg( )

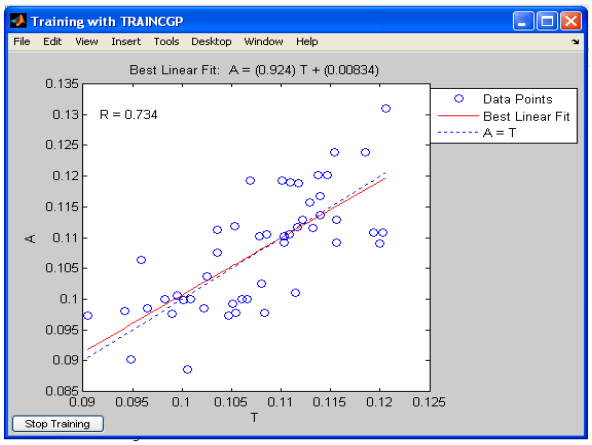

Fig. 5.40: Predicted strength vs. target strength for Exp - I using traincgp()

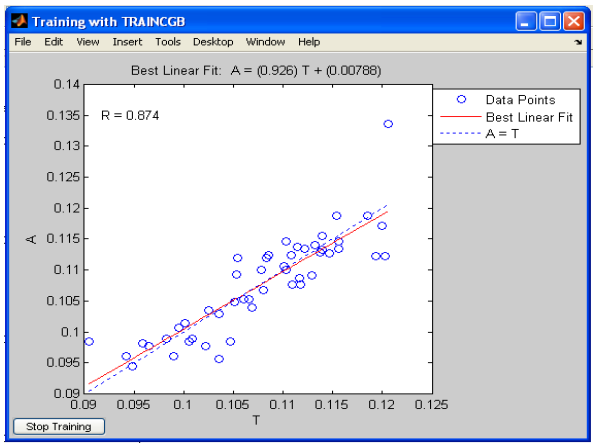

Fig. 5.41: Predicted strength vs. target strength for Exp - I using traincgb()

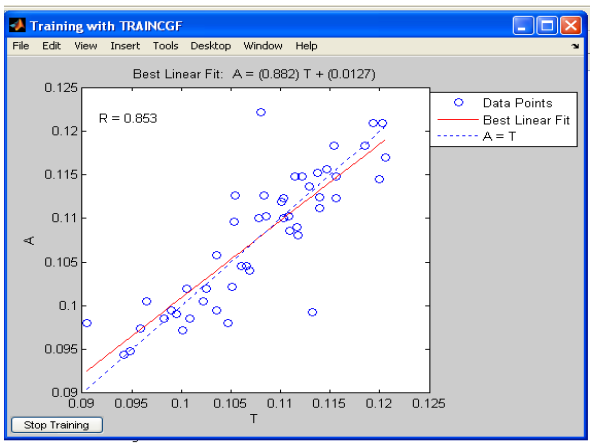

Fig. 5.42: Predicted strength vs. target strength for Exp - I using traincgf()

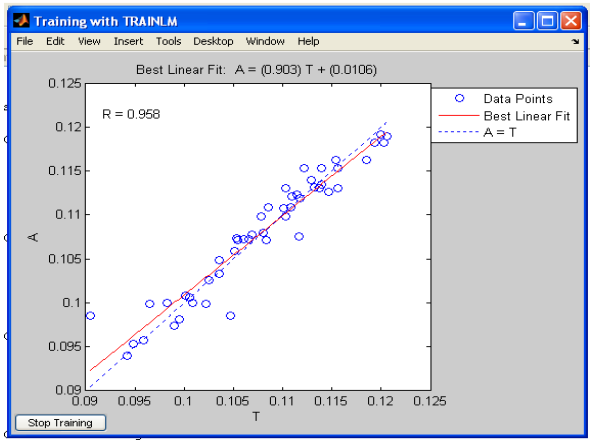

Fig. 5.43: Predicted strength vs. target strength for Exp - I using trainlm()

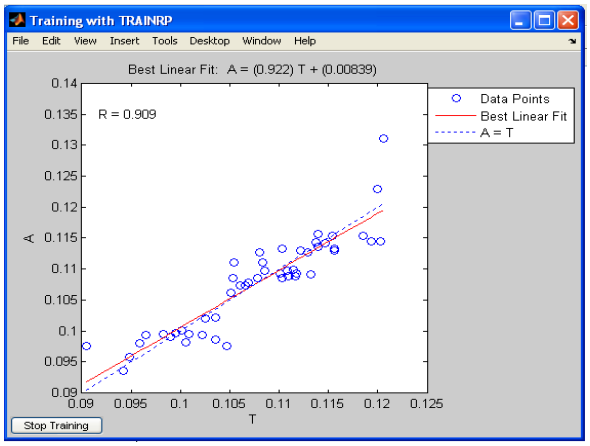

Fig. 5.44: Predicted strength vs. target strength for Exp - I using trainrp()

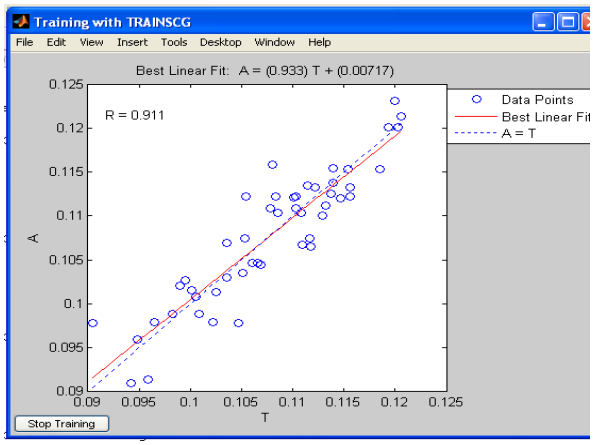

Fig. 5.45: Predicted strength vs. target strength for Exp - I using trainscg()

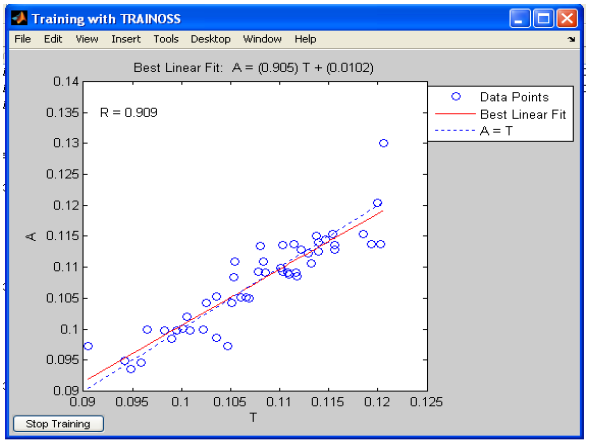

Fig. 5.46: Predicted strength vs. target strength for Exp - I using trainoss()

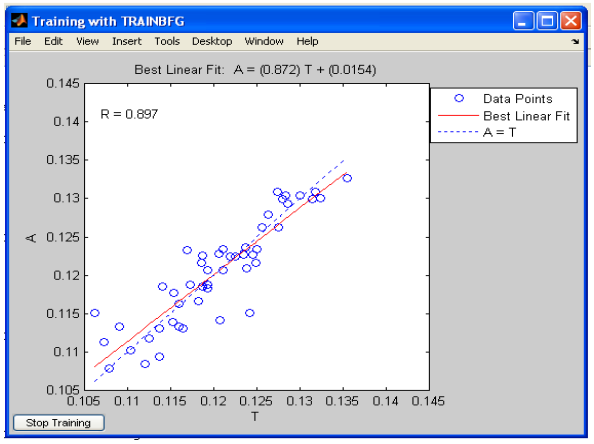

Fig. 5.47: Predicted strength vs. target strength for Exp - II using trainbfg()

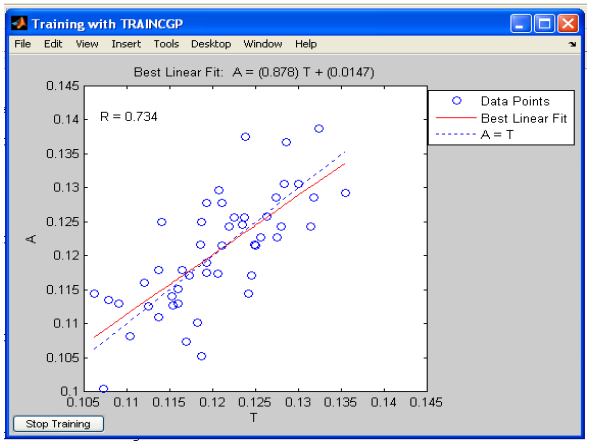

Fig. 5.48: Predicted strength vs. target strength for Exp - II using traincgp()

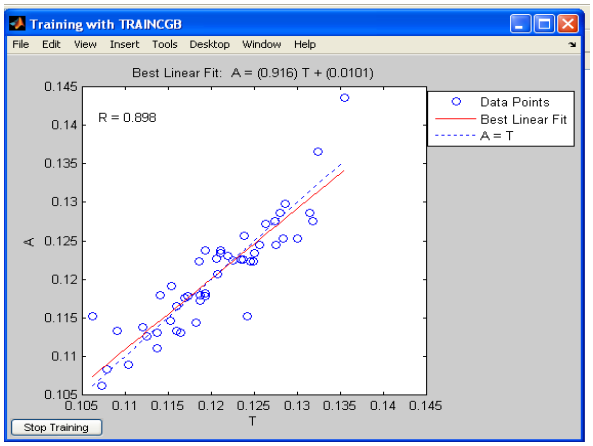

Fig. 5.49: Predicted strength vs. target strength for Exp - II using traincgb()

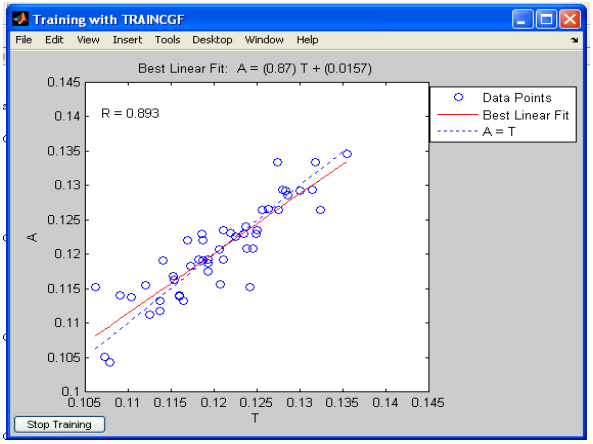

Fig. 5.50: Predicted strength vs. target strength for Exp - II using traincgf()

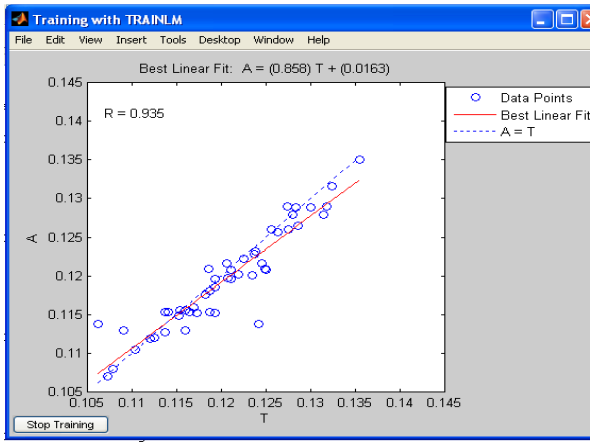

Fig. 5.51: Predicted strength vs. target strength for Exp - II using trainlm()

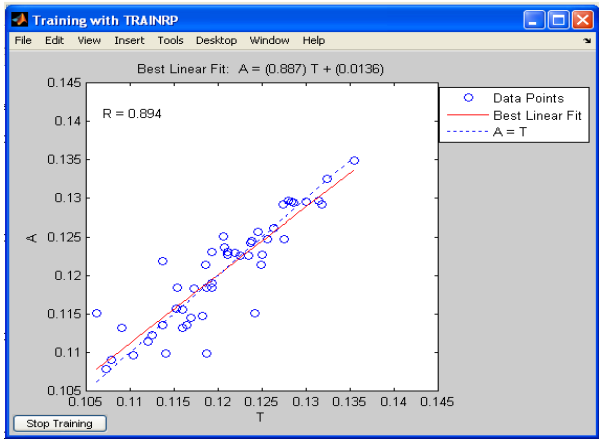

Fig. 5.52: Predicted strength vs. target strength for Exp - II using trainrp()

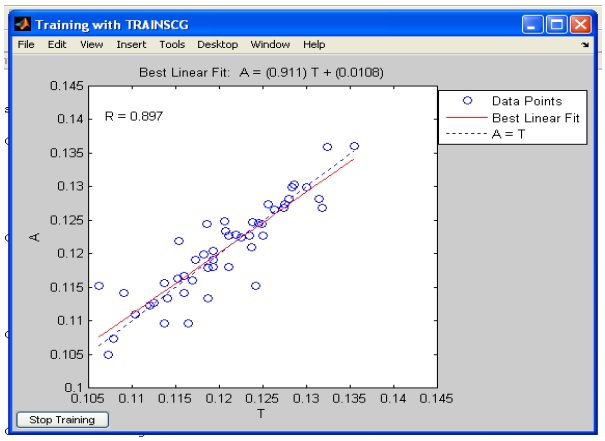

Fig. 5.53: Predicted strength vs. target strength for Exp - II using trainscg()

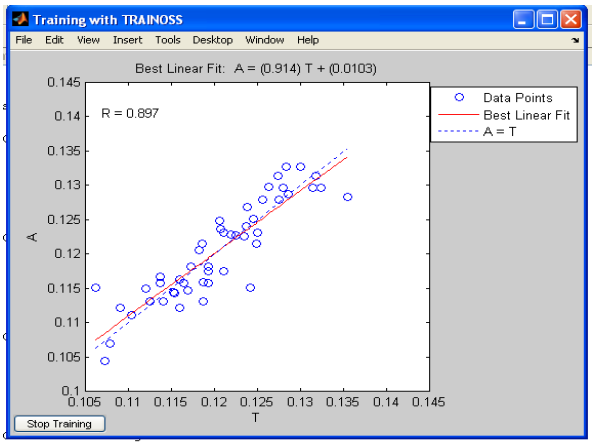

Fig. 5.54: Predicted strength vs. target strength for Exp - II using trainoss()

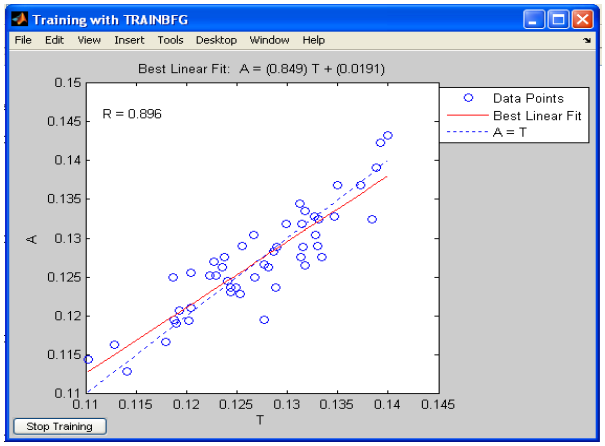

Fig. 5.55: Predicted strength vs. target strength for Exp - III using trainbfg()

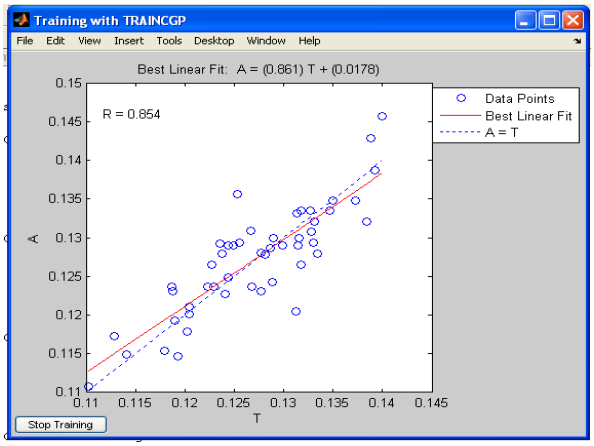

Fig. 5.56: Predicted strength vs. target strength for Exp - III using traincgp()

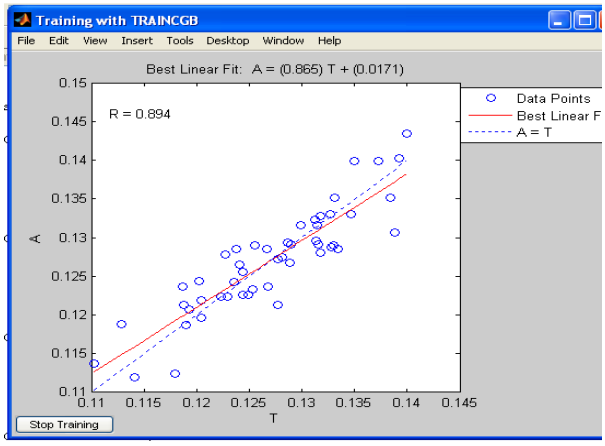

Fig. 5.57: Predicted strength vs. target strength for Exp - III using traincgb()

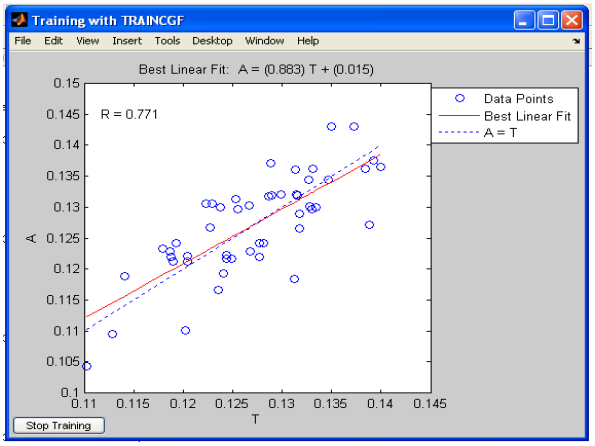

Fig. 5.58: Predicted strength vs. target strength for Exp - III using traincgf()

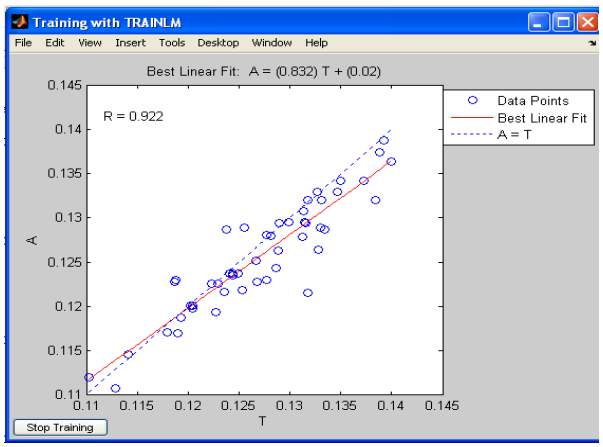

Fig. 5.59: Predicted strength vs. target strength for Exp - III using trainlm()

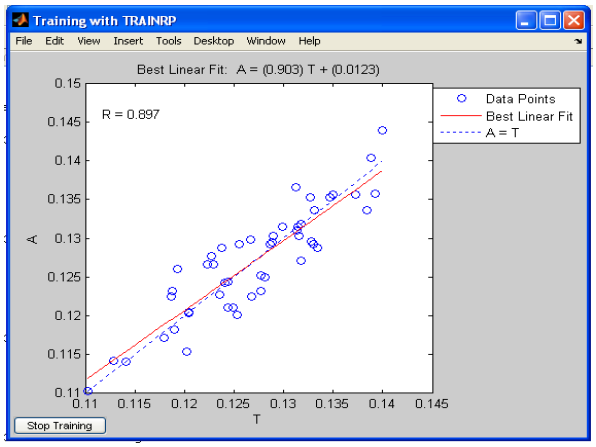

Fig. 5.60: Predicted strength vs. target strength for Exp - III using trainrp()

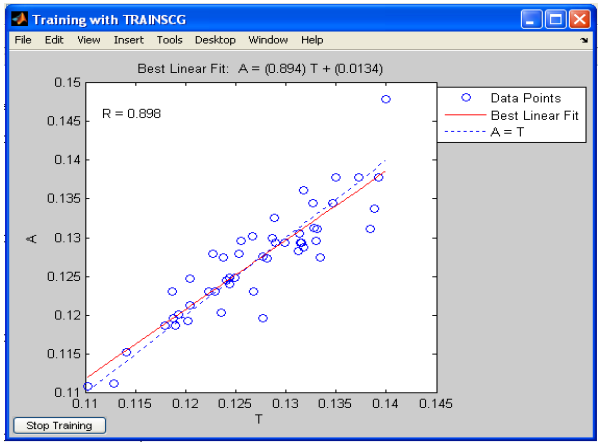

Fig. 5.61: Predicted strength vs. target strength for Exp - III using trainscg()

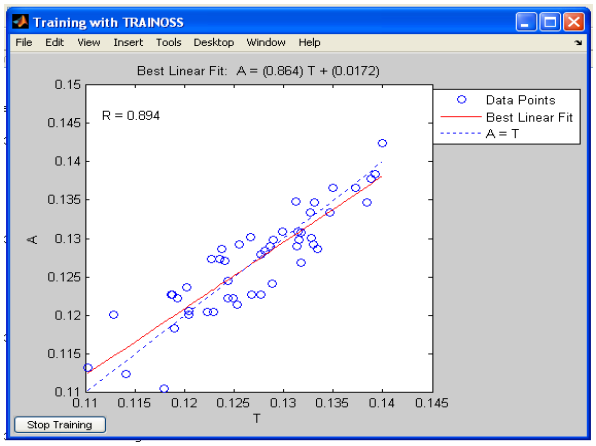

Fig. 5.62: Predicted strength vs. target strength for Exp - III using trainoss()

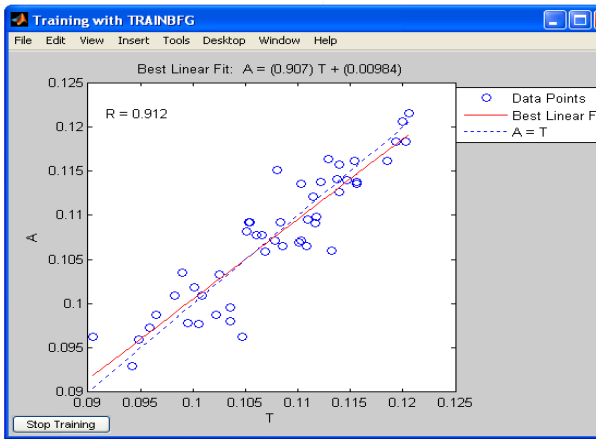

Fig. 5.63: Predicted strength vs. target strength for Exp - IV using trainbfg()

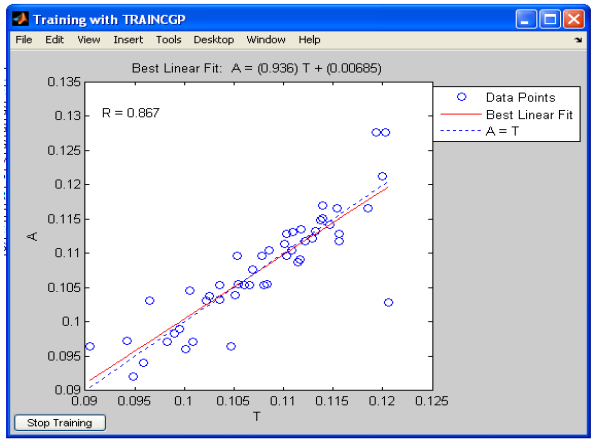

Fig. 5.64: Predicted strength vs. target strength for Exp - IV using traincgp()

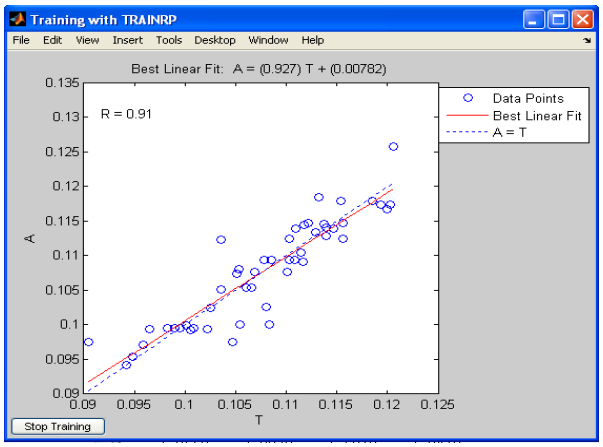

Fig. 5.65: Predicted strength vs. target strength for Exp - IV using trainrp()

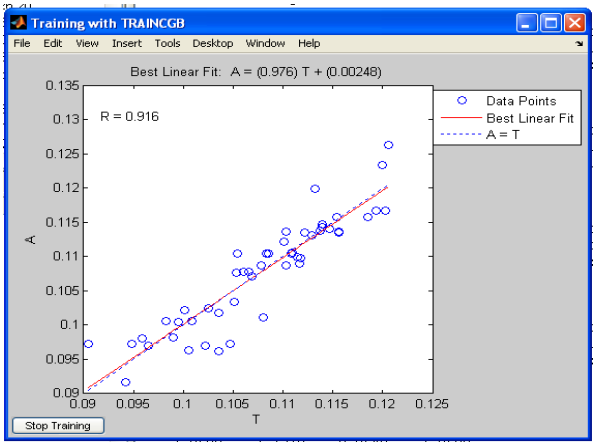

Fig. 5.66: Predicted strength vs. target strength for Exp - IV using traincgb()

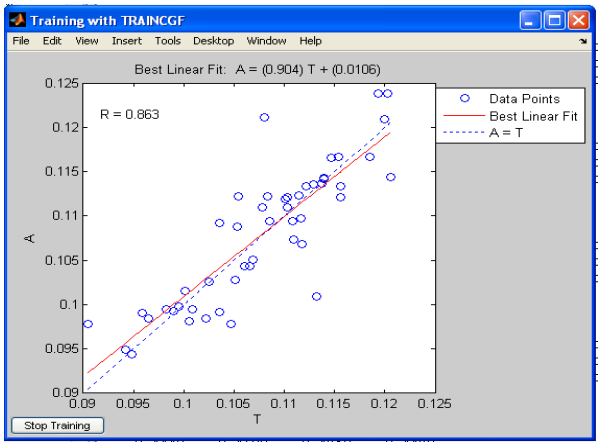

Fig. 5.67: Predicted strength vs. target strength for Exp - IV using traincgf()

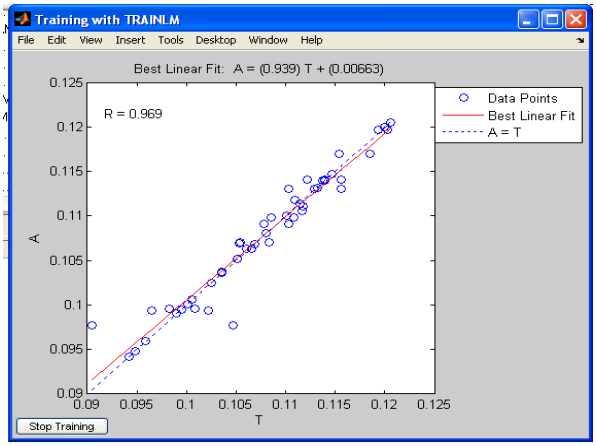

Fig. 5.68: Predicted strength vs. target strength for Exp - IV using trainlm()

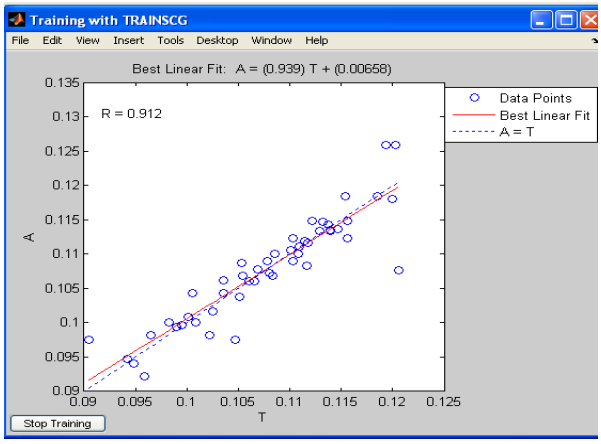

Fig. 5.69: Predicted strength vs. target strength for Exp - IV using trainscg()

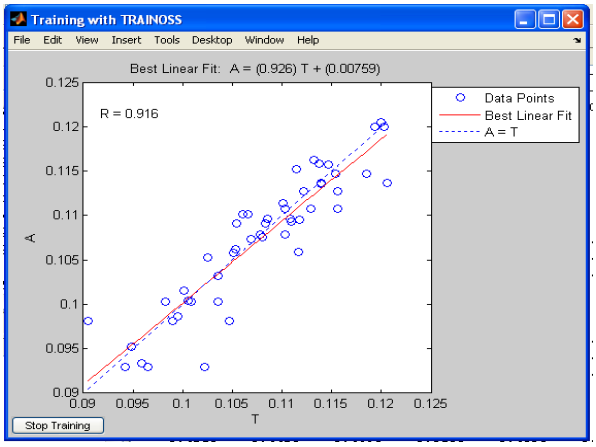

Fig. 5.70: Predicted strength vs. target strength for Exp - IV using trainoss()

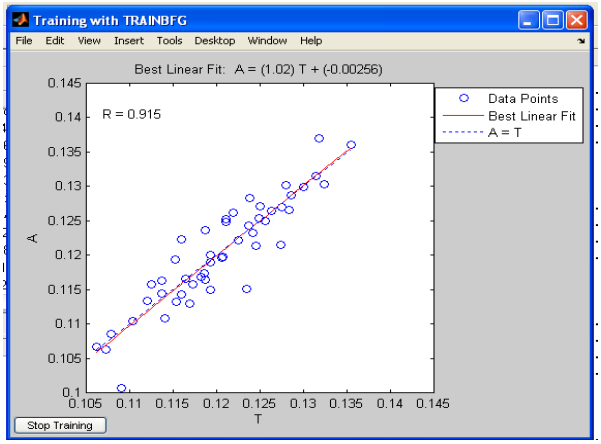

Fig. 5.71: Predicted strength vs. target strength for Exp - V using trainbfg()

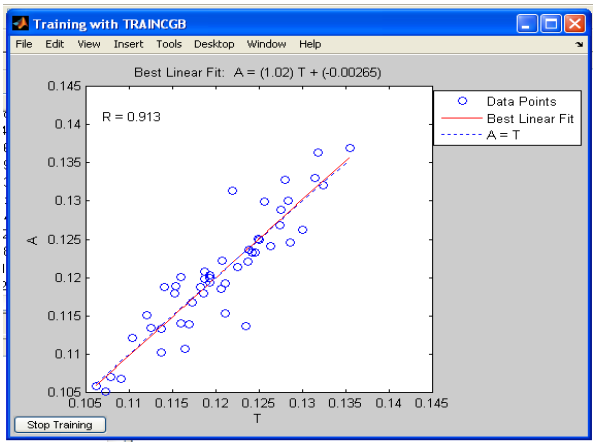

Fig. 5.72: Predicted strength vs. target strength for Exp - V using traincgb()

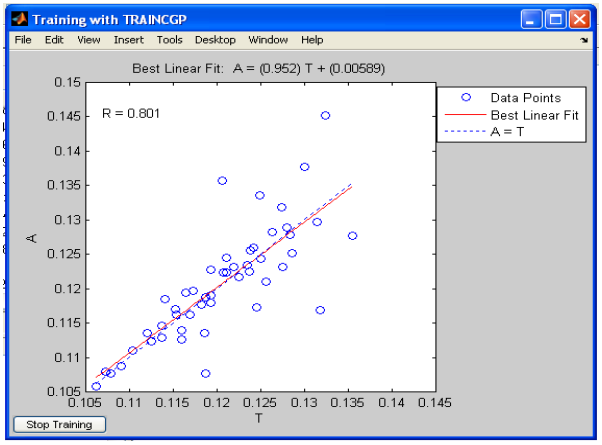

Fig. 5.73: Predicted strength vs. target strength for Exp - V using traincgp()

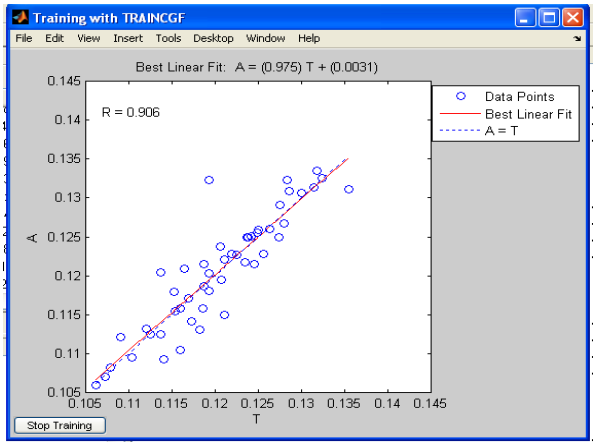

Fig. 5.74: Predicted strength vs. target strength for Exp - V using traincgf()

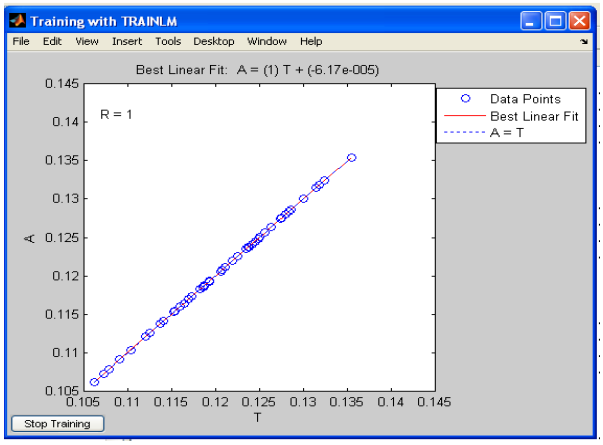

Fig. 5.75: Predicted strength vs. target strength for Exp - V using trainlm()

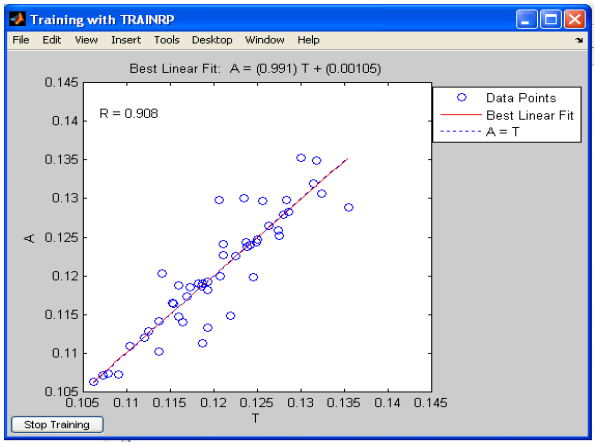

Fig. 5.76: Predicted strength vs. target strength for Exp - V using trainrp()

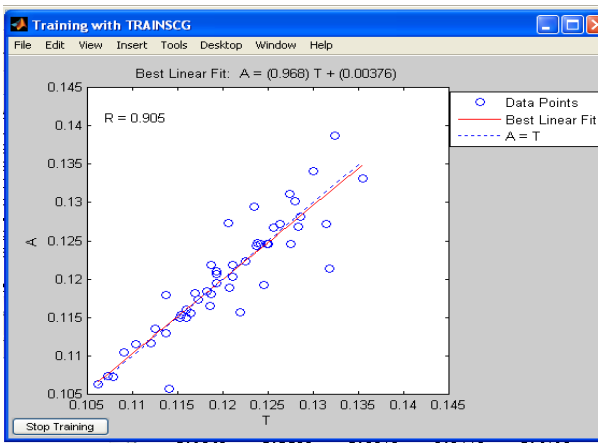

Fig. 5.77: Predicted strength vs. target strength for Exp - V using trainscg()

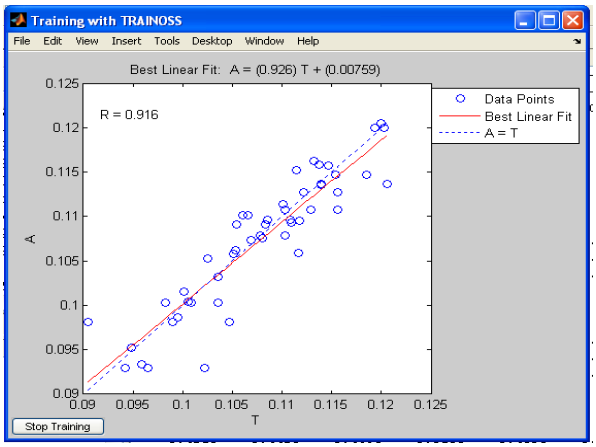

Fig. 5.78: Predicted strength vs. target strength for Exp - V using trainoss()

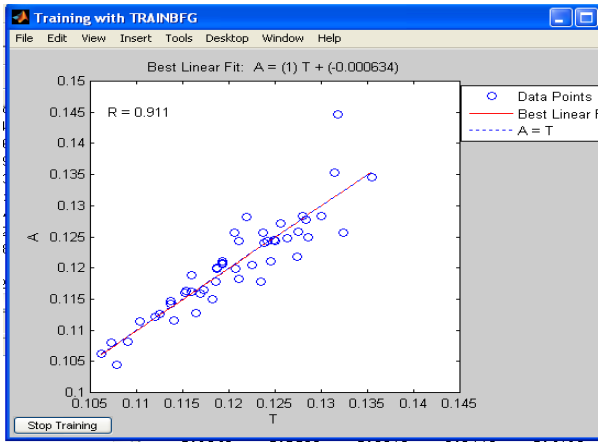

Fig. 5.79: Predicted strength vs. target strength for Exp - VI using trainbfg()

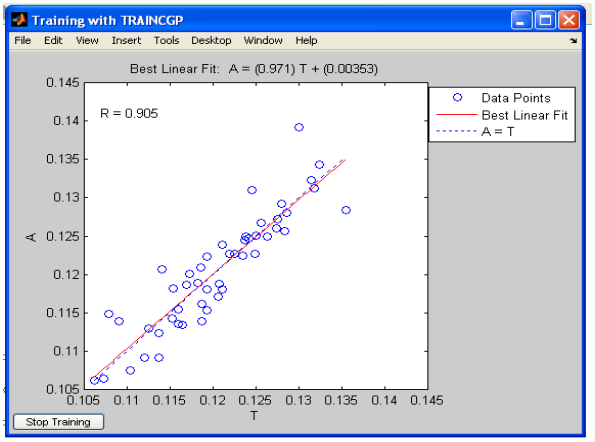

Fig. 5.80: Predicted strength vs. target strength for Exp - VI using traincgp()

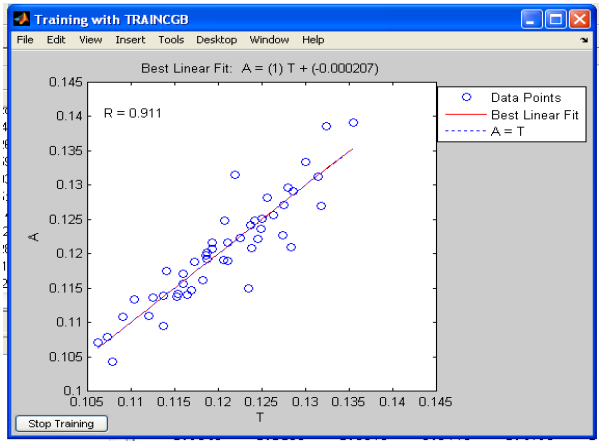

Fig. 5.81: Predicted strength vs. target strength for Exp - VI using traincgb()

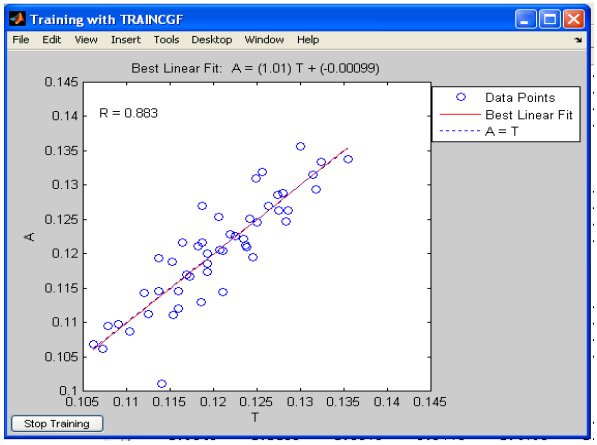

Fig. 5.82: Predicted strength vs. target strength for Exp - VI using traincgf()

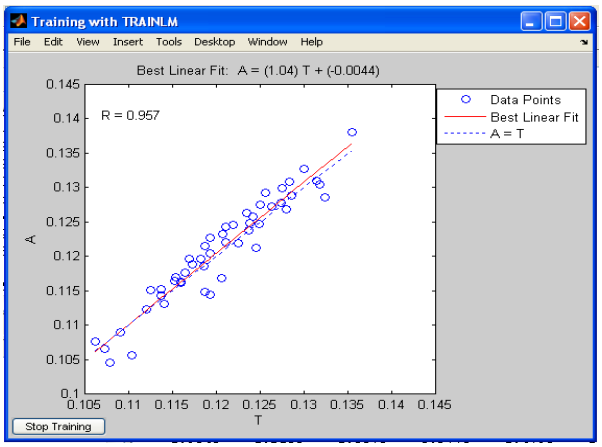

Fig. 5.83: Predicted strength vs. target strength for Exp - VI using trainlm()

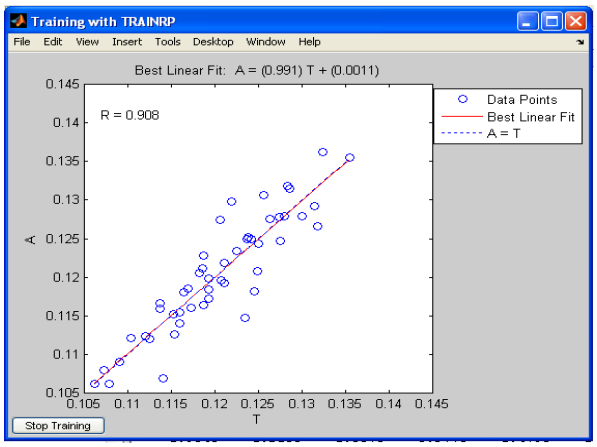

Fig. 5.84: Predicted strength vs. target strength for Exp - VI using trainrp()

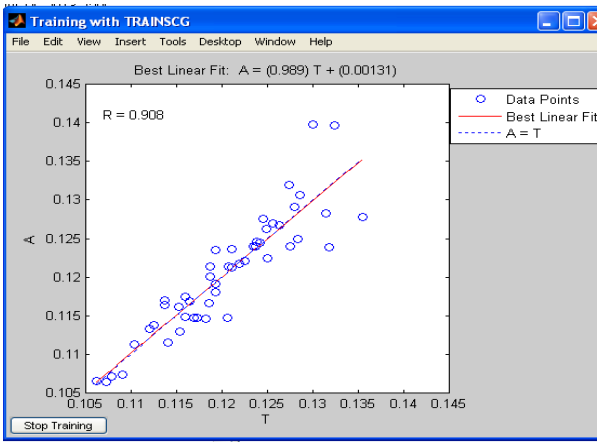

Fig. 5.85: Predicted strength vs. target strength for Exp - VI using trainscg()

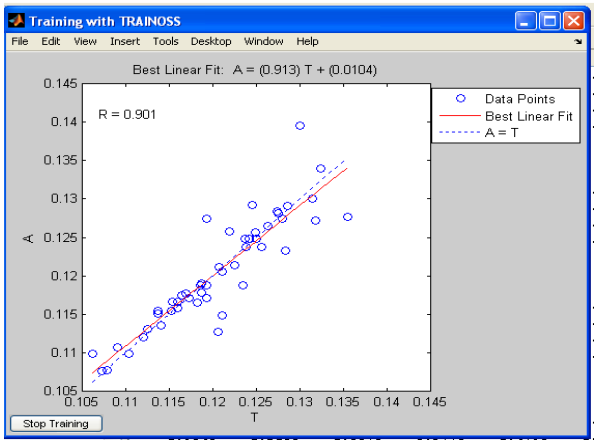

Fig. 5.86: Predicted strength vs. target strength for Exp - VI using trainoss()

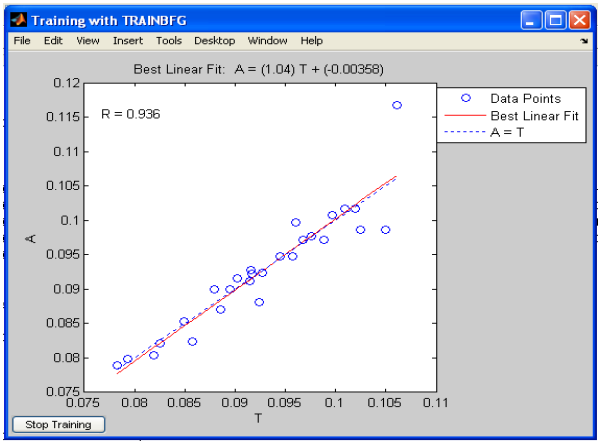

Fig. 5.87: Predicted strength vs. target strength for Exp - VII using trainbfg()

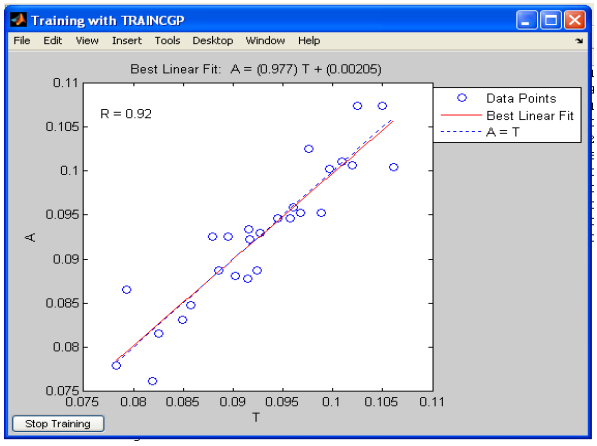

Fig. 5.88: Predicted strength vs. target strength for Exp - VII using traincgp()

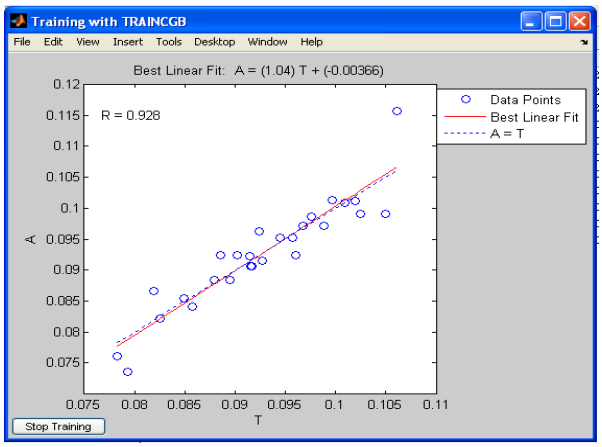

Fig. 5.89: Predicted strength vs. target strength for Exp - VII using traincgb()

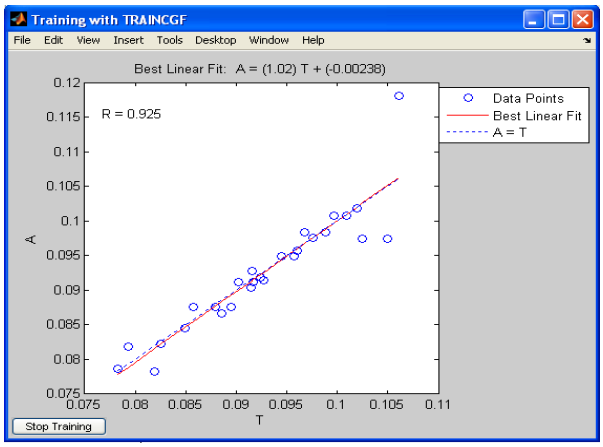

Fig. 5.90: Predicted strength vs. target strength for Exp - VII using traincgf()

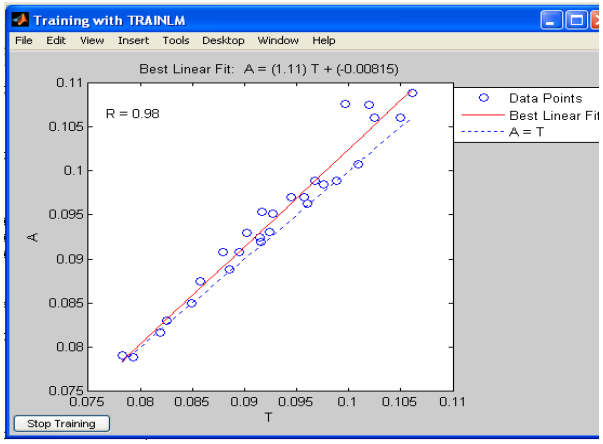

Fig. 5.91: Predicted strength vs. target strength for Exp - VII using trainlm()

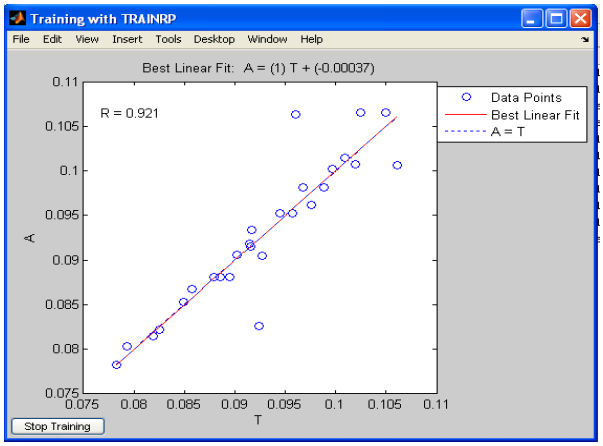

Fig. 5.92: Predicted strength vs. target strength for Exp - VII using trainrp()

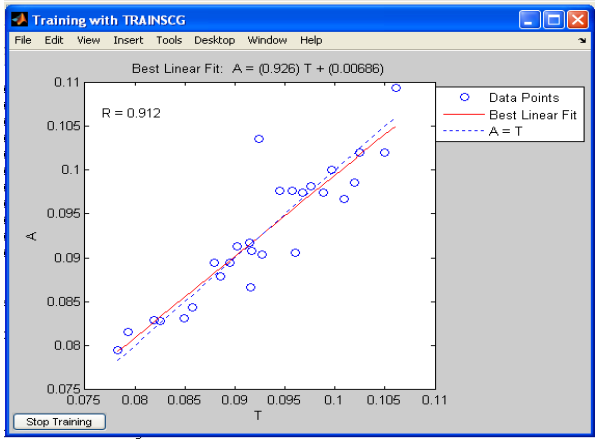

Fig. 5.93: Predicted strength vs. target strength for Exp - VII using trainscg()

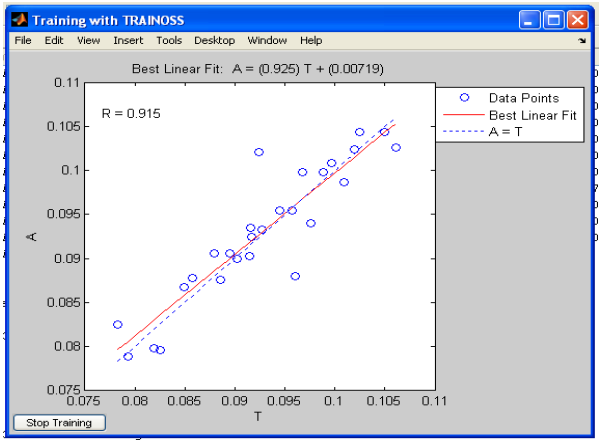

Fig. 5.94: Predicted strength vs. target strength for Exp - VII using trainoss()

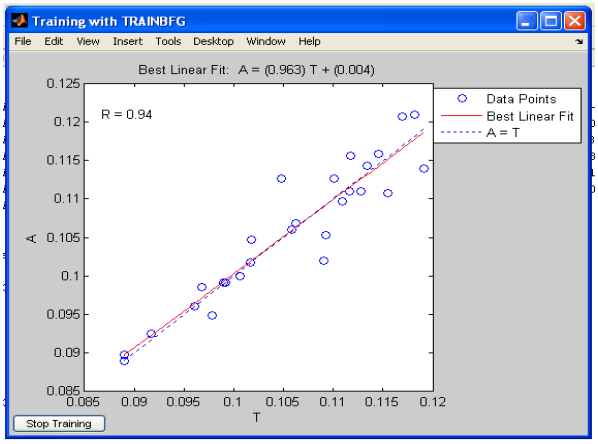

Fig. 5.95: Predicted strength vs. target strength for Exp - VIII using trainbfg()

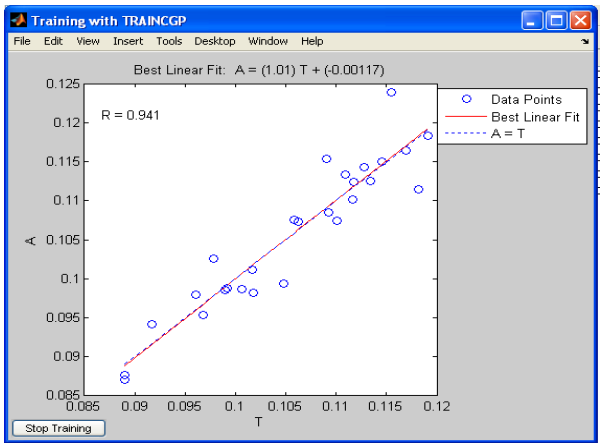

Fig. 5.96: Predicted strength vs. target strength for Exp - VIII using traincgp()

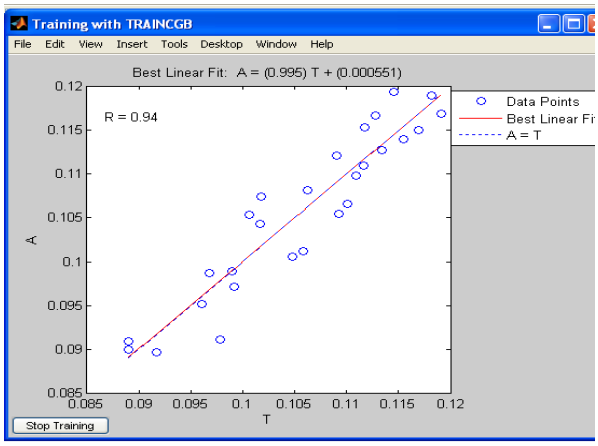

Fig. 5.97: Predicted strength vs. target strength for Exp - VIII using traincgb()

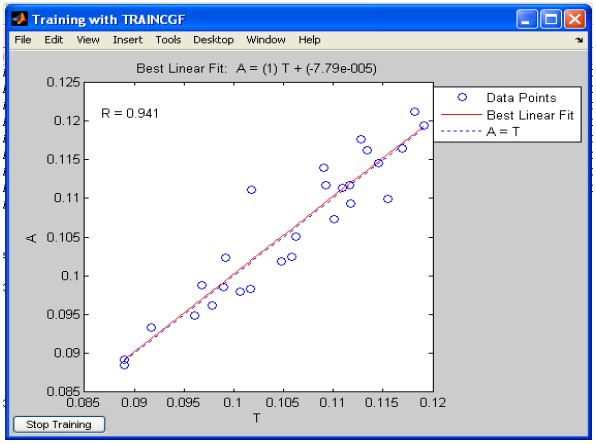

Fig. 5.98: Predicted strength vs. target strength for Exp - VIII using traincgf()

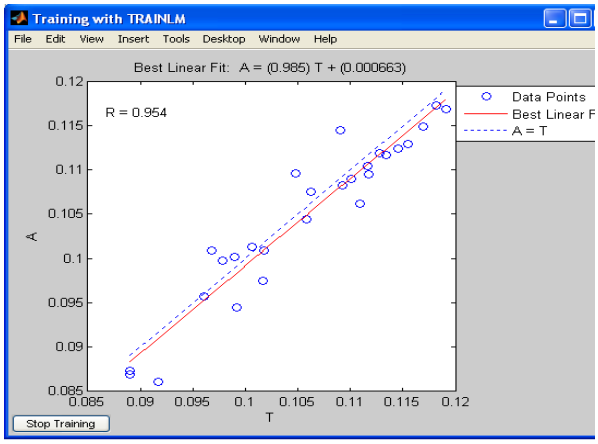

Fig. 5.99: Predicted strength vs. target strength for Exp - VIII using trainlm()

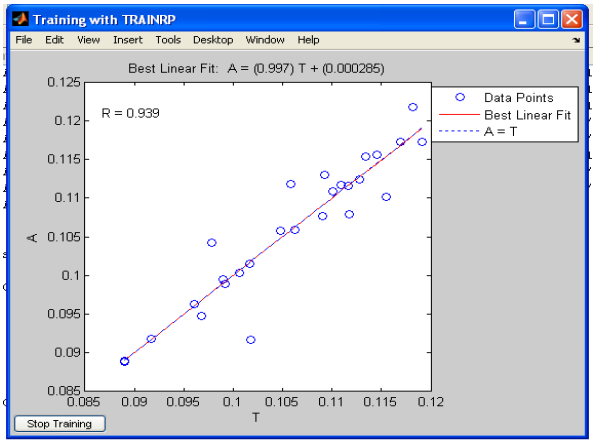

Fig. 5.100: Predicted strength vs. target strength for Exp - VIII using trainrp()

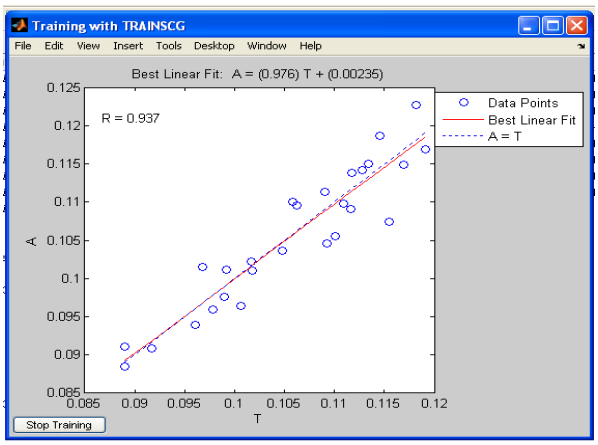

Fig. 5.101: Predicted strength vs. target strength for Exp - VIII using trainscg()

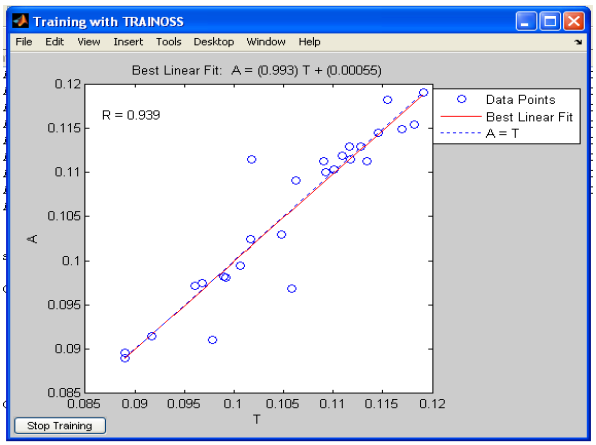

Fig. 5.102: Predicted strength vs. target strength for Exp - VIII using trainoss()

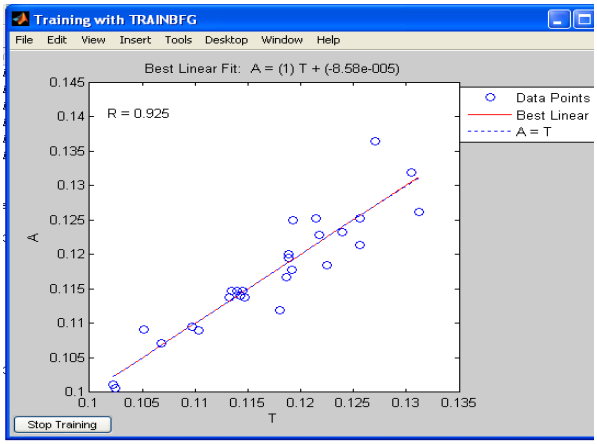

Fig. 5.103: Predicted strength vs. target strength for Exp - IX using trainbfg()

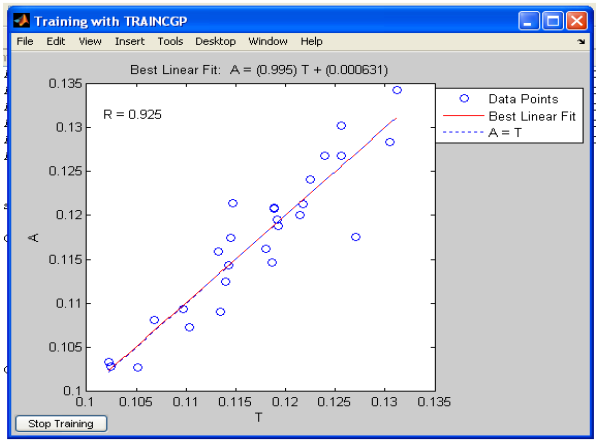

Fig. 5.104: Predicted strength vs. target strength for Exp - IX using traincgp()

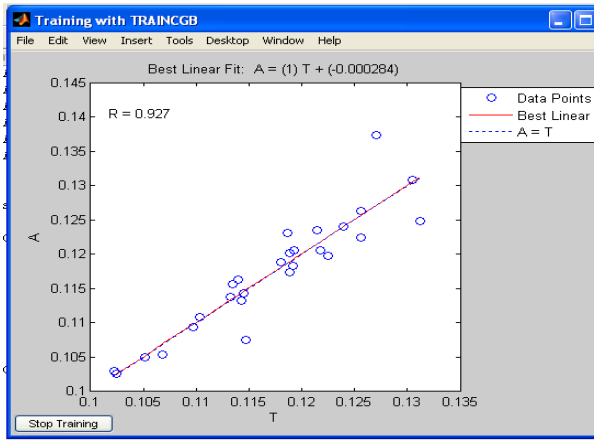

Fig. 5.105: Predicted strength vs. target strength for Exp - IX using traincgb()

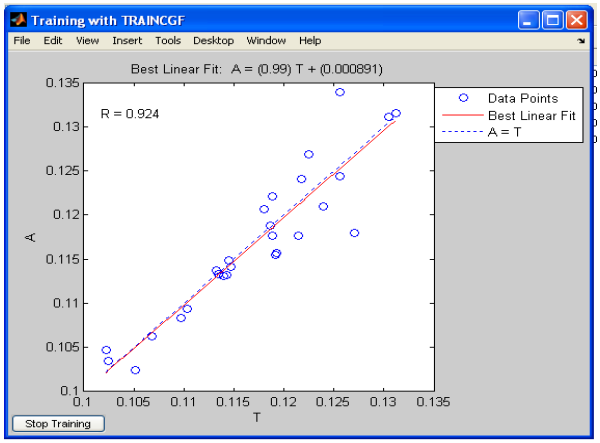

Fig. 5.106: Predicted strength vs. target strength for Exp - IX using traincgf()

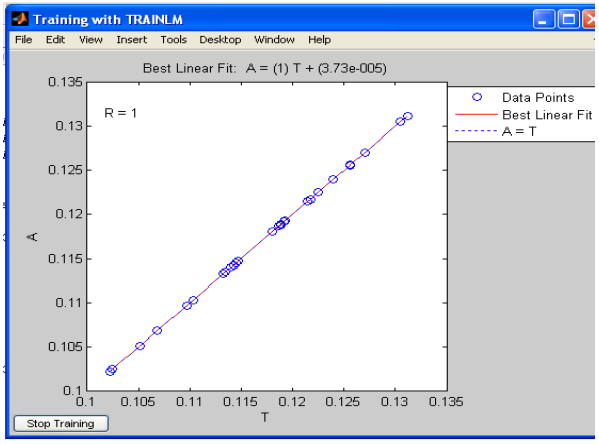

Fig. 5.107: Predicted strength vs. target strength for Exp - IX using trainlm()

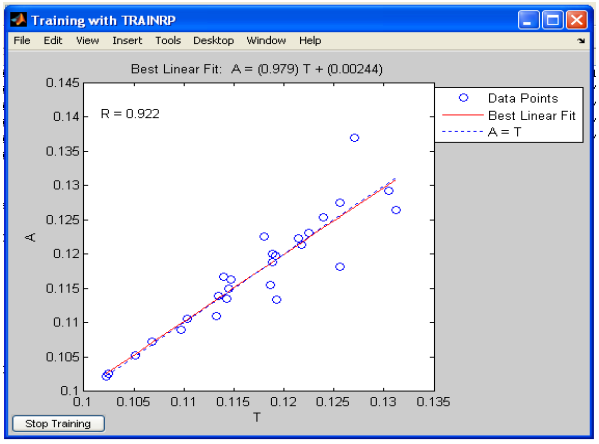

Fig. 5.108: Predicted strength vs. target strength for Exp - IX using trainrp()

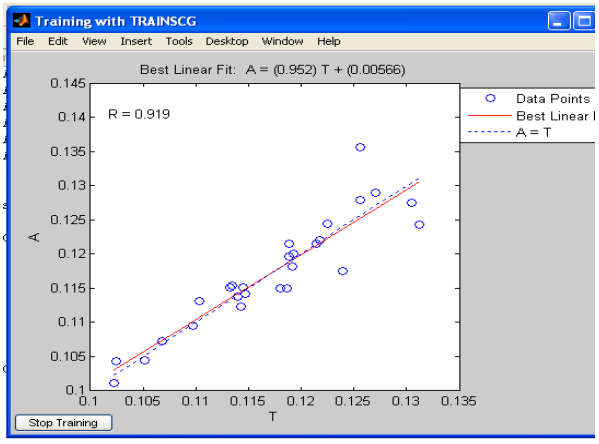

Fig. 5.109: Predicted strength vs. target strength for Exp - IX using trainscg()

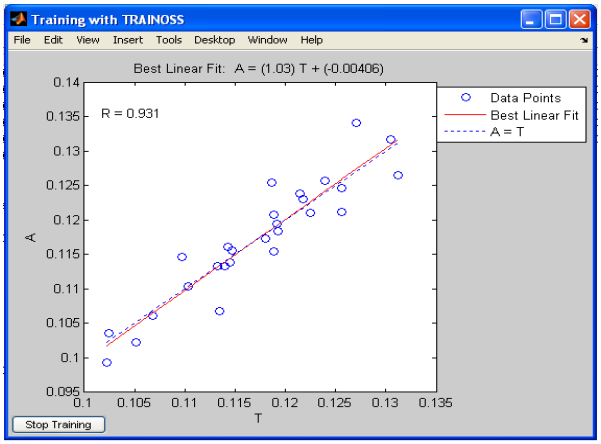

Fig. 5.110: Predicted strength vs. target strength for Exp - IX using trainoss()

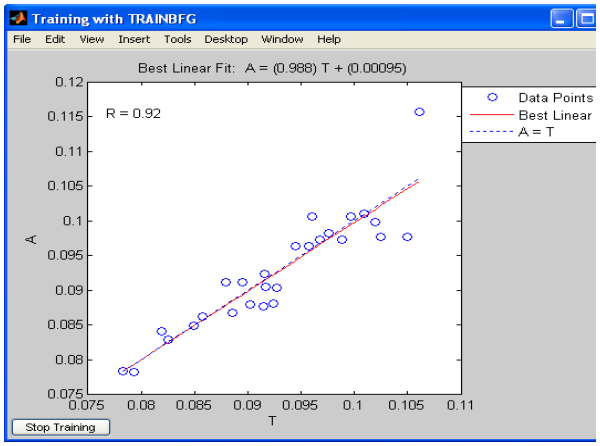

Fig. 5.111: Predicted strength vs. target strength for Exp - X using trainbfg()

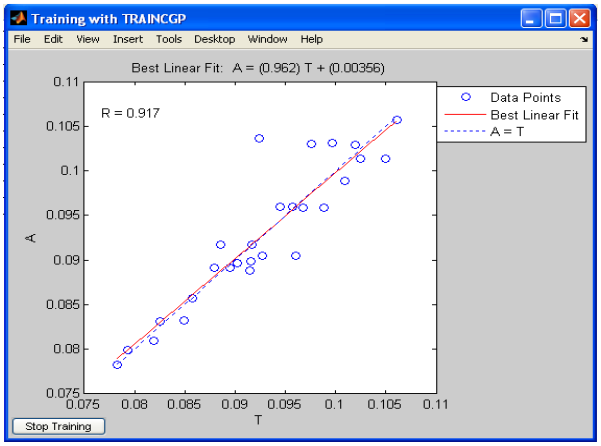

Fig. 5.112: Predicted strength vs. target strength for Exp - X using traincgp()

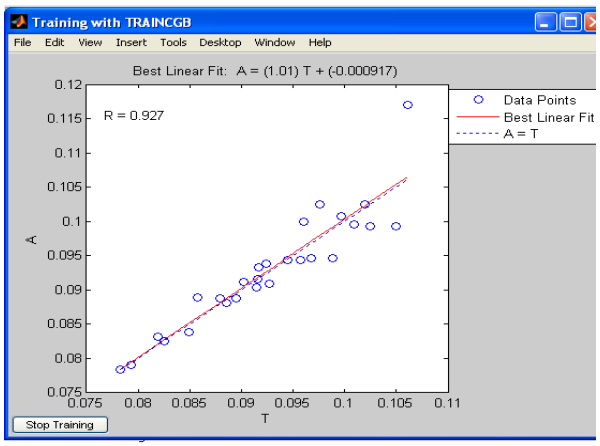

Fig. 5.113: Predicted strength vs. target strength for Exp - X using traincgb()

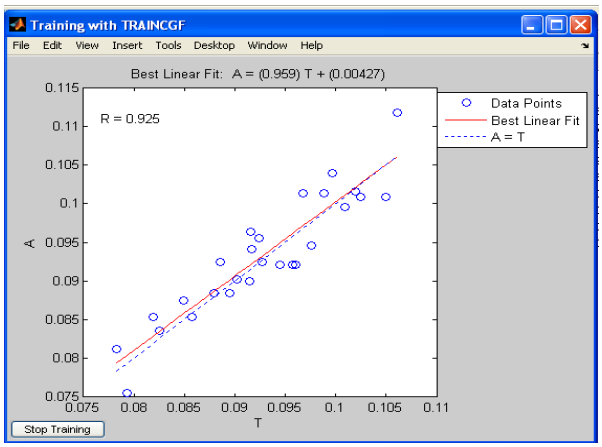

Fig. 5.114: Predicted strength vs. target strength for Exp - X using traincgf()

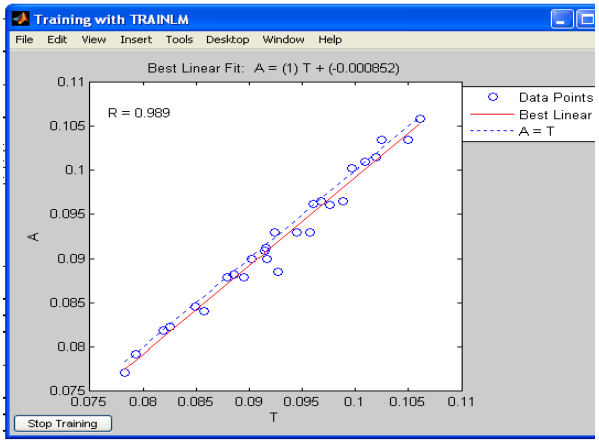

Fig. 5.115: Predicted strength vs. target strength for Exp - X using trainlm()

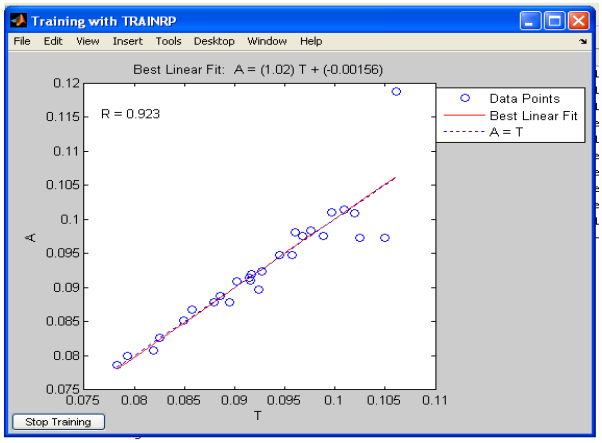

Fig. 5.116: Predicted strength vs. target strength for Exp - X using trainrp()

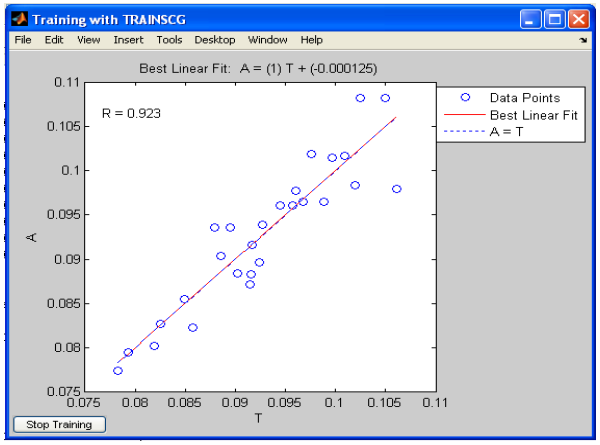

Fig. 5.117: Predicted strength vs. target strength for Exp - X using trainscg()

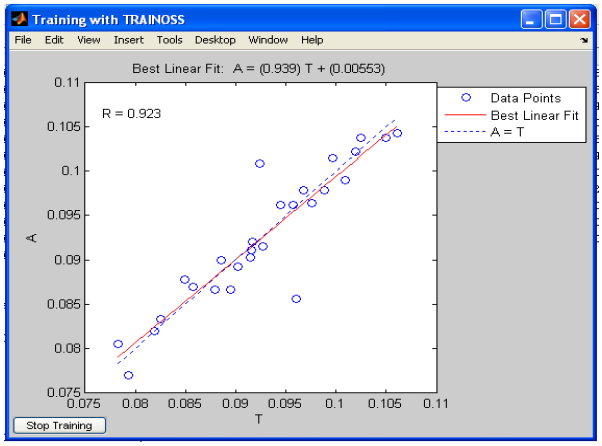

Fig. 5.118: Predicted strength vs. target strength for Exp - X using trainoss()

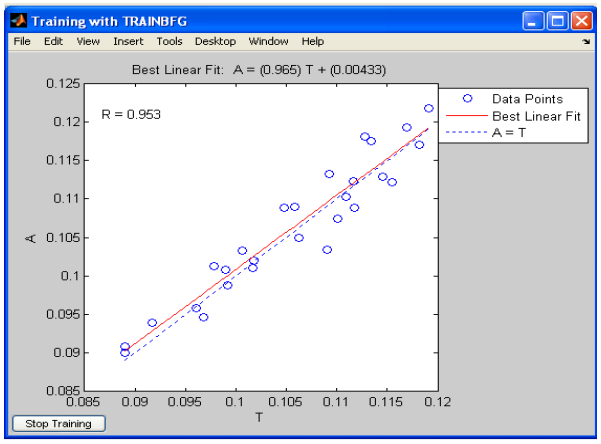

Fig. 5.119: Predicted strength vs. target strength for Exp - XI using trainbfg()

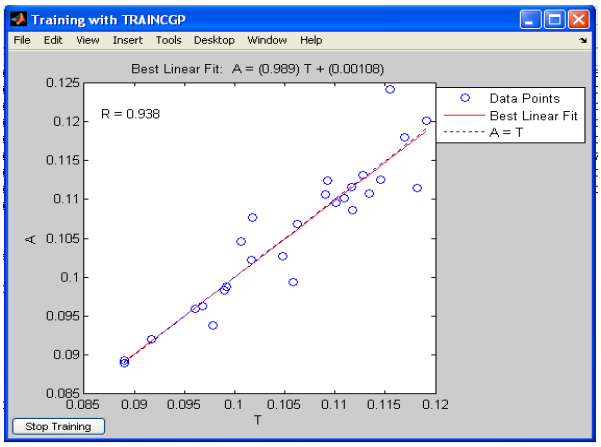

Fig. 5.120: Predicted strength vs. target strength for Exp - XI using traincgp()

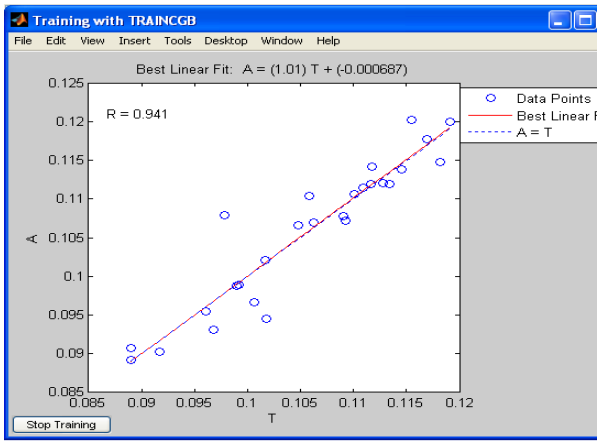

Fig. 5.121: Predicted strength vs. target strength for Exp - XI using traincgb()

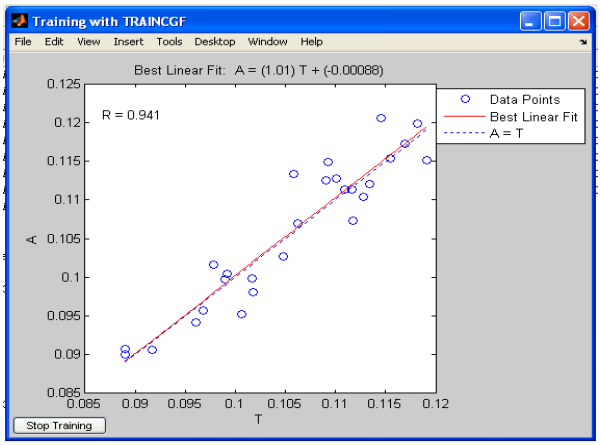

Fig. 5.122: Predicted strength vs. target strength for Exp - XI using traincgf()

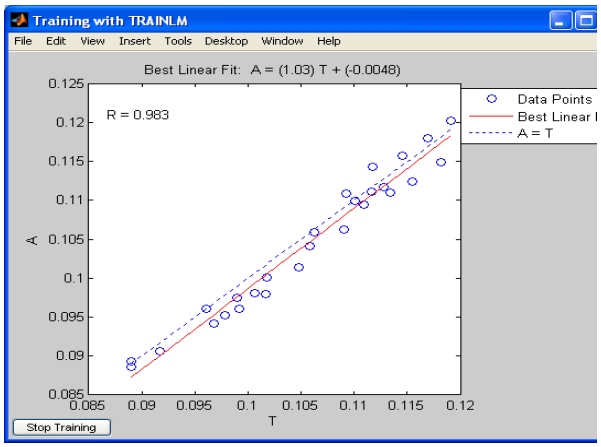

Fig. 5.123: Predicted strength vs. target strength for Exp - XI using trainlm()

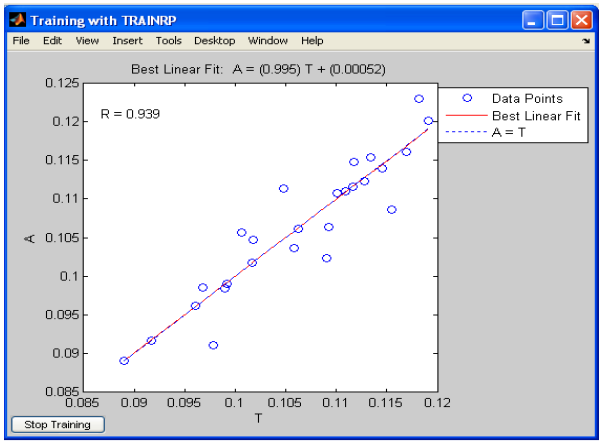

Fig. 5.124: Predicted strength vs. target strength for Exp - XI using trainrp()

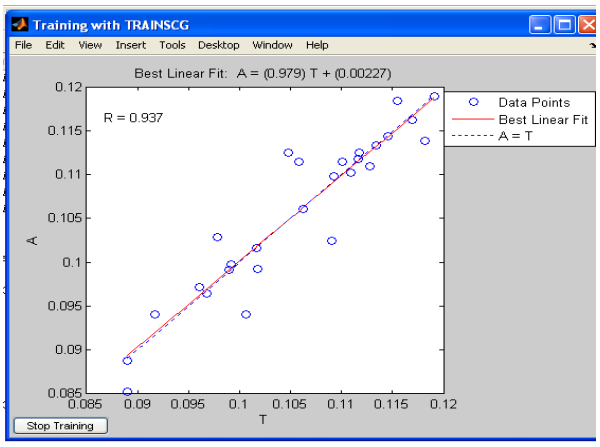

Fig. 5.125: Predicted strength vs. target strength for Exp - XI using trainscg()

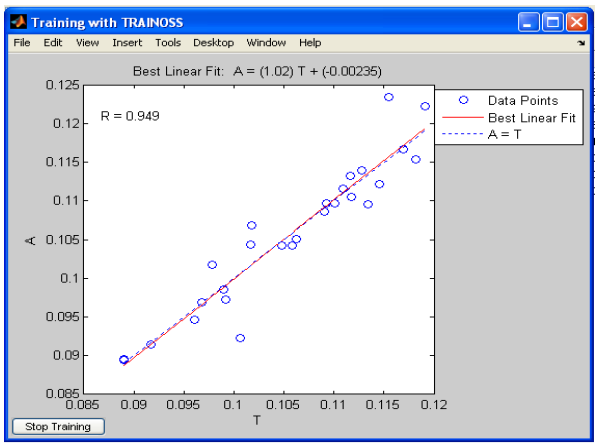

Fig. 5.126: Predicted strength vs. target strength for Exp - XI using trainoss()

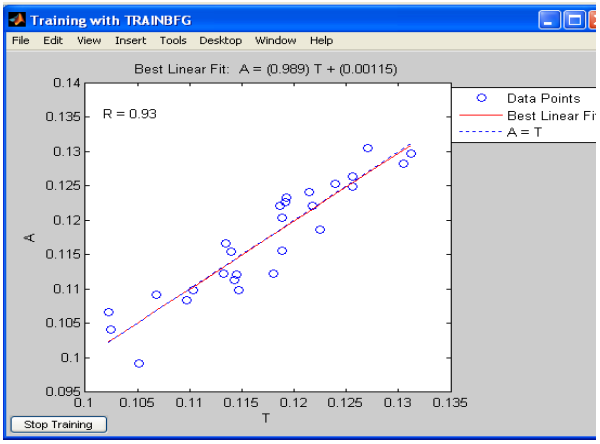

Fig. 5.127: Predicted strength vs. target strength for Exp - XII using trainbfg()

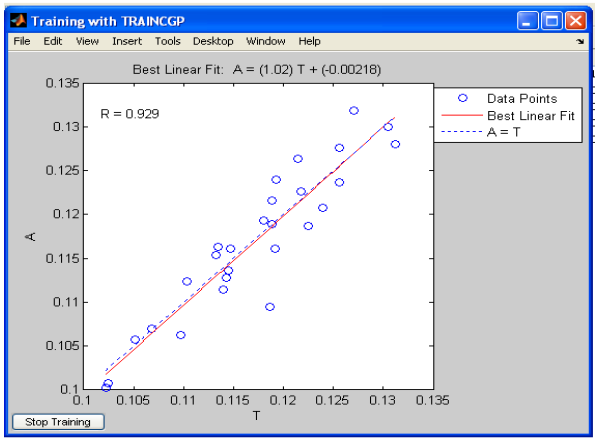

Fig. 5.128: Predicted strength vs. target strength for Exp - XII using traincgp()

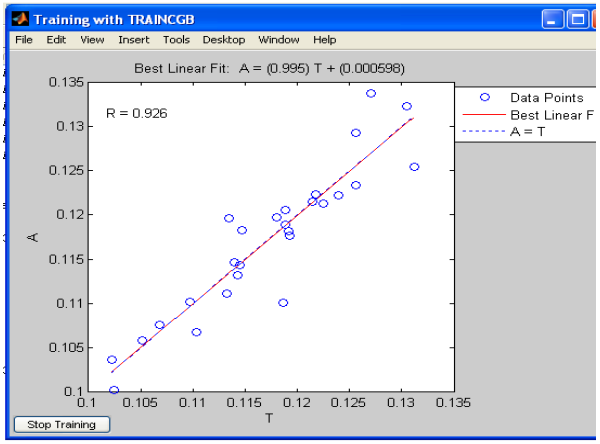

Fig. 5.129: Predicted strength vs. target strength for Exp - XII using traincgb()

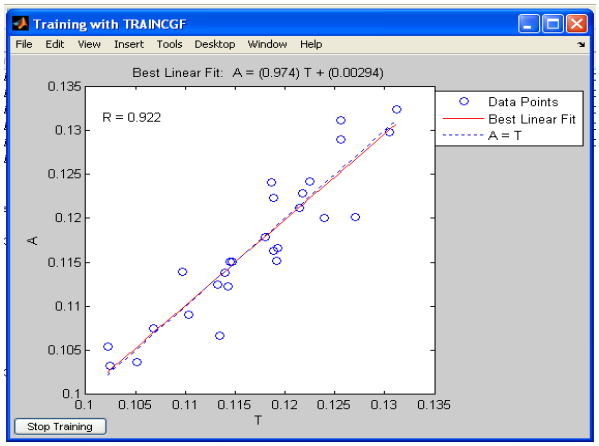

Fig. 5.130: Predicted strength vs. target strength for Exp - XII using traincgf()

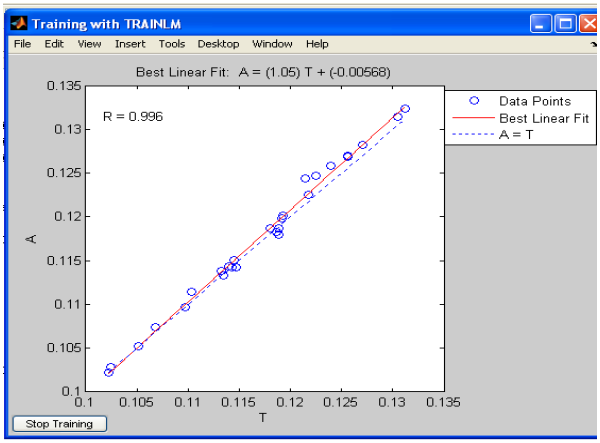

Fig. 5.131: Predicted strength vs. target strength for Exp - XII using trainlm()

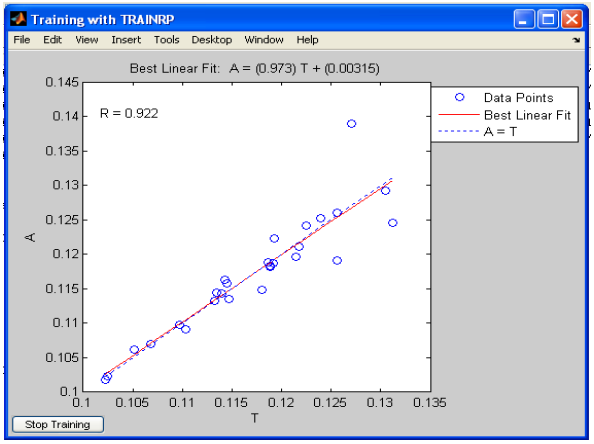

Fig. 5.132: Predicted strength vs. target strength for Exp - XII using trainrp()

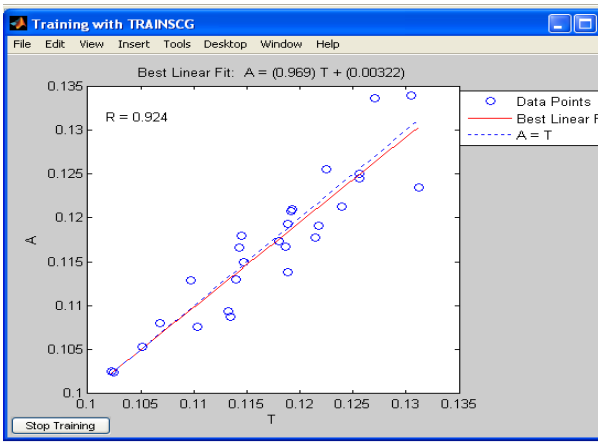

Fig. 5.133: Predicted strength vs. target strength for Exp - XII using trainscg()

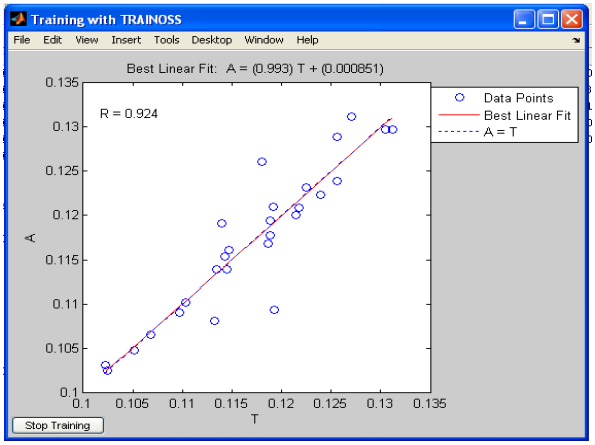

Fig. 5.134: Predicted strength vs. target strength for Exp XII using trainoss()

5.3.3 Conclusions

In this section, neural network models have been proposed to predict the compressive strength of concrete. In the development of this model, eight training functions have experimentally been investigated for network optimization. Also, we empirically investigated different architectural parameters such as the number of hidden neurons (*hn*), learning rate (*lr*), AFs, performance goal (*goal*) and epochs for the optimal values of parameters of neural network models. For the two datasets, experiments in this section, trainlm() training function, with tan-sigmoid as AF gives the best results. Therefore, it is deduced that the best training algorithm for the prediction of compressive strength of concrete is 'Levenberg-Marquardt' algorithm with tan-sigmoid as the AF for hidden layer. We have used linear activation function for output layer in all experiments.

5.4 Development of genetic programming model

The dataset used for GP model development is same as in Section 5.2 for the development of neural network models. The control parameters as given in Koza (1992) have been investigated in this study. For the development of GP models, the initial population size is 49 tuples for Dataset - 1 and 27 tuples for Dataset - 2. The population size (*mu*) and the number of children produced (*lambda*) have been taken 100 and 150, respectively. As, larger is the number of generations, greater are the chances of evolving a solution, the number of generations has been taken as 100000. The values for the parameters, namely, crossover rate and mutation rate have been selected as 0.70 and 0.05, respectively, after experimentations. The values of other parameters, *i.e.,* training percentage, selection method and tournament size of substitution are taken as 75, 'tournament' and 3, respectively for the development of GP model. Function set ('+, -, *, /') has been taken for both the datasets. The architectural details of the selected GP model are given in Table 5.40.

Table 5.40: Architecture of GP model

Parameters	Values	Description
Initial Population Size	Dataset - 1	Dataset - 1 is of 49 tuples in total and in this dataset, *cm* is not replaced by FA.
	Dataset - 2	Dataset - 2 is of 27 tuples in total and in this dataset, *cm* is replaced by FA.
Number of input parameters	4 (*cm*, *w*, *fa*, *ca)*	When output is for f_{c28}, then number of input parameters is 4.
	5 (*cm*, *w*, *fa*, *ca,* f_{c28}*)*	When output is for f_{c56}, the number of input parameters is 5 as f_{c28} is taken as input.
	6 (*cm*, *w*, *fa*, *ca,* f_{c28}, f_{c56})	When output is for f_{c91}, the number of input parameters is 6 as $f_{c28,}$ f_{c56} are also taken as input.
Function Set	+, -, *, /, sqrt	Set of functions used
Training percentage	75	---
Selection Method	Tournament	---
Tournament size of replacement	3	---

Maximum Generations	100000	Maximum number of iterations
Crossover	0.7	Probability of crossover
Mutation	0.05	Probability of mutation
mu	100	Population size
lambda	150	No. of children produced
Objectives	R^2	Co-efficient of determination
	RMSE	Root mean square error

5.4.1 Prediction of compressive strength of concrete without FA and with 15% FA

In the GP model development, the addition is chosen as the linking utility. The prediction equations generated using the GP model are listed below ((5.1) to (5.6)). Here, f_{c28}, f_{c56} and f_{c91} are the compressive strengths of concrete at curing ages of 28, 56 and 91 days, respectively, for Dataset - 1. For Dataset - 2, f'_{c28}, f'_{c56} and f'_{c91} are the compressive strengths of concrete at curing ages of 28, 56 and 91 days, respectively.

$$f_{c28} = \sqrt{r} * \left(\frac{\frac{q}{0.6618} + \sqrt{r}}{\sqrt{q}} + \sqrt{q} + \left(2 * (q + r + 1.3201)\right) - \frac{1}{\left(p * \left(1 + (2 * q)\right)\right)^{0.125}} \right) \tag{5.1}$$

$$f_{c56} = \left(\left(\frac{p*\left(1+\sqrt{2p}\right)*f_{c28}}{p*\left((2*p)+(3*f_{c28})+\sqrt{f_{c28}}\right)} \right) + p + 0.4887 \right) * f_{c28} \tag{5.2}$$

$$f_{c91} = \sqrt{\left(f_{c56} * \left(\sqrt{\sqrt{\left(\frac{f_{c28}}{p^3}\right) + \sqrt{p * (r - q)} + f_{c28}} + f_{c28}} \right) * (f_{c28} + p^2) \right) + \left(\frac{f_{c28}}{p}\right)} \tag{5.3}$$

$$f'_{c28} = \frac{\sqrt{r}}{\left(q * \sqrt{q * \left((2 * q) + r\right)} \right) + (2 * r) + (3 * q) + (4 * p) + \sqrt{r} + \sqrt{q}} \tag{5.4}$$

$$f'_{c56} = \sqrt{\frac{-0.0643+f'_{c28}+(r)^{0.3125}}{\left(\left((r)^{0.375}*\sqrt{\frac{p}{f'_{c28}}+\sqrt{q}\pm 0.0678}\right)+f'_{c28}\right)^{0.25}}} \tag{5.5}$$

$$f'_{c91} = 2.23 * p^{0.625} * f'_{c28} \tag{5.6}$$

where, p = w/cm, $q = fa/cm$ and $r = ca/cm$.

5.4.2 Results and discussions

This section presents the results obtained from the approach and quantitative assessment of the model's predictive abilities. The results of the experiments performed are tabulated in Table 5.41

Table 5.41: Results obtained from GP model

Genetic Programming (GP) Model		Training results	
Dataset	**Prediction variables**	**R^2 (%)**	**RMSE**
Dataset - 1	f_{c28}	77.4	0.0107
	f_{c56}	99.9	0.0055
	f_{c91}	99.9	0.0064
Dataset - 2	f'_{c28}	93.8	0.0142
	f'_{c56}	94.5	0.0091
	f'_{c91}	96.7	0.0068

It can be observed from Table 5.41 that for Dataset - 1, the value of R^2 is 99.9% for the prediction of f_{c56} and f_{c91}. However, for the prediction of f_{c28}, the value of R^2 obtained is 77.4%, which is 2.6% less than the suggested good model fit (R^2 should be more than or equal to 80.0%). In case of prediction of compressive strength for Dataset - 2, the value of R^2 obtained is 93.8%, 94.5% and 96.7% for the prediction of f'_{c28}, f'_{c56} and f'_{c91}, respectively. This indicates that the GP model provides highest results for the prediction of f_{c56} and f_{c91} (Dataset - 1) and for the predicton of f'_{c91} (Dataset - 2). One can also infer that GP model is acceptable for all the curing ages except for the prediction of f_{c28}.

5.4.3 Conclusions

The objective of this study was to explore the applicability of GP models for compressive strength prediction. We can argue that if a model has the value of R^2 greater or equal to 80.0%, there is a well-built correlation between predicted and measured values for the data availables in the dataset. It has been observed that the value of R^2 is greater than 80.0% for all the curing days, except for the prediction of f_{c28}. As an outcome, we can say that GP may serve as a predictive model for the prediction of compressive strength of concrete.

5.5 Validation of neural network and GP models

To further test the efficacy and reliability of the models, the in-situ compressive strength data for the prediction of f_{c28} (as provided in Namyong *et al.*, 2004) has been used in this study. Namyong *et al.* (2004) have proposed the regression equations for prediction of in-situ compressive strength of concerete and for this purpose they have used the information of mixture proportions of ready-mixed concrete and test results of compressive strength from construction sites. In their study, they have used 1442 compressive strength test results obtained from the specimens having 68 different kinds of mixtures with specified compressive strength of 18~27 MPa, *w/cm* of 0.39~0.62, maximum aggregate size of 25 mm. This data has been included in Table 5.42. In this study, Namyong *et al.,* (2004) in-situ data has been used for the validation of suggested models for prediction of compressive strength of concrete. Fig. 5.135 shows the results of the comparison of models for the in-situ dataset of compressive strength. It has been clearly observed that neural network is more reliable and provides more accurate prediction for the in-situ dataset for concrete.

Table 5.42: Comparison of neural network model with GP model using f_{c28} data from literature

w/cm	*fa/cm*	*ca/cm*	f_{c28} target (in MPa-m^3/kg)	f_{c28} predicted by neural network model (in MPa-m^3/kg)	f_{c28} predicted by GP model (in MPa-m^3/kg)
0.602	3.228	3.114	0.073	0.077	0.090
0.597	3.010	2.792	0.079	0.081	0.091
0.606	3.066	2.848	0.076	0.079	0.090
0.575	3.193	2.867	0.087	0.089	0.089
0.603	2.870	2.737	0.076	0.079	0.092

0.615	2.948	2.780	0.073	0.076	0.091
0.595	2.657	3.155	0.073	0.076	0.097
0.500	2.701	2.872	0.106	0.107	0.095
0.500	2.701	2.872	0.106	0.107	0.095
0.478	2.188	2.685	0.073	0.076	0.101
0.494	2.383	2.539	0.085	0.088	0.097
0.524	2.468	2.738	0.085	0.087	0.097
0.445	2.183	2.695	0.074	0.078	0.101
0.447	2.308	2.561	0.078	0.081	0.099
0.486	2.534	2.592	0.092	0.094	0.096
0.489	2.341	2.397	0.083	0.085	0.097
0.500	2.588	2.670	0.093	0.095	0.095
0.486	2.534	2.592	0.091	0.094	0.096
0.489	2.341	2.397	0.082	0.085	0.097
0.500	2.588	2.670	0.092	0.094	0.095
0.489	2.341	2.397	0.089	0.091	0.097
0.500	2.588	2.670	0.099	0.101	0.095
0.497	2.376	2.401	0.087	0.089	0.097
0.448	2.088	2.235	0.076	0.079	0.100
0.453	1.990	2.365	0.076	0.079	0.103
0.466	1.931	2.496	0.076	0.079	0.105
0.500	2.297	2.729	0.085	0.088	0.100
0.470	2.000	2.473	0.069	0.072	0.103
0.473	2.000	2.473	0.065	0.069	0.103
0.480	2.115	2.413	0.073	0.076	0.101
0.484	2.135	2.386	0.077	0.080	0.100
0.484	2.164	2.444	0.077	0.080	0.100
0.478	2.109	2.395	0.072	0.075	0.101
0.457	2.209	2.363	0.075	0.079	0.099
0.468	2.403	2.373	0.074	0.077	0.096
0.467	2.295	2.251	0.072	0.075	0.097
0.442	2.284	2.376	0.077	0.080	0.098
0.478	2.131	2.509	0.069	0.072	0.101
0.451	2.139	2.467	0.074	0.077	0.101
0.484	2.135	2.442	0.073	0.076	0.101
0.459	1.880	2.439	0.078	0.081	0.105
0.486	2.100	2.483	0.075	0.078	0.102
0.467	2.295	2.251	0.073	0.077	0.097
0.442	2.284	2.376	0.075	0.078	0.098

0.457	2.253	2.319	0.074	0.077	0.098
0.478	2.206	2.681	0.077	0.080	0.101
0.459	1.772	2.299	0.075	0.078	0.106
0.460	1.781	2.314	0.075	0.078	0.106
0.428	1.862	2.202	0.071	0.075	0.104
0.409	1.584	2.171	0.068	0.072	0.109
0.409	1.564	2.184	0.069	0.072	0.109
0.420	1.844	2.039	0.068	0.072	0.103
0.441	1.915	2.080	0.071	0.075	0.102
0.436	1.869	2.190	0.073	0.076	0.104
0.443	1.889	2.230	0.073	0.076	0.104
0.443	1.915	2.235	0.075	0.078	0.103
0.410	1.978	2.283	0.081	0.084	0.102
0.425	1.917	2.101	0.073	0.076	0.102
0.443	1.896	2.315	0.071	0.074	0.104
0.484	1.941	2.170	0.076	0.079	0.102
0.452	1.976	2.133	0.071	0.075	0.101
0.443	1.889	2.179	0.083	0.086	0.103
0.443	1.896	2.213	0.078	0.081	0.103
0.394	1.700	2.170	0.066	0.069	0.107
0.438	1.853	2.293	0.074	0.077	0.105
0.425	1.917	2.101	0.075	0.078	0.102
0.415	1.853	2.220	0.075	0.078	0.104
0.438	1.969	2.485	0.078	0.081	0.104

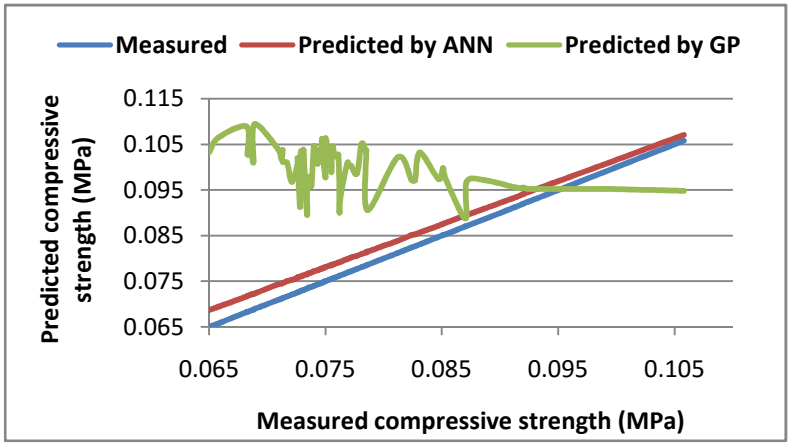

Fig. 5.135: Validation of models (neural network model and GP model) for f_{c28} with in-situ dataset

5.6 Comparison of neural network model with other predictive models

5.6.1 Comparison with GP model

The objective of the present study was to explore the applicability of suggested models, *i.e.,* neural network model and GP model for the prediction of compressive strength of concrete. This section presents the comparative investigation of results obtained from these approaches and quantitative assessment of the models' predictive abilities. In Section 5.4.3, the training function trainlm() has given best results for the prediction of compressive strength of concrete; therefore, this function alongwith tan-sigmoid as AF has been used for evaluating the prediction accuracy parameters for neural network model. The results, as presented in Table 5.43, gives the values of R^2 and RMSE for prediction of compressive strength of concrete for Dataset - 1 and Dataset - 2. From these results, it can be observed that, for all the curing days, for both of the Datasets, the value of R^2 is greater than or equal to 99.6% except for f_{c28} for Dataset - 1. The low values of RMSE at different curing ages also indicate that the model can predict compressive strength with high reliability. Also, it can be seen that the neural network model, with trainlm() training function, achieves the result in just a few epochs, which clearly indicates that the time taken for the prediction is also very less.

Table 5.43: Results obtained from neural network model

Neural network model		**Training results**		
Dataset	**Prediction variables**	**Epochs taken**	R^2 **(%)**	**RMSE**
Dataset - 1	f_{c28}	04	89.8	6.9762e-006
	f_{c56}	05	99.8	1.2712e-007
	f_{c91}	03	100	7.3640e-009
Dataset - 2	f'_{c28}	05	99.6	3.8809e-007
	f'_{c56}	04	100	3.6873e-009
	f'_{c91}	04	100	2.2181e-010

The prediction equations ((5.1) - (5.6)) generated using GP model in Section 5.4 are further used for the prediction of compressive strength of concrete and the results have been presented in

Table 5.41. The values of R^2 and RMSE for prediction of concrete compressive strength for Dataset - 1 and Dataset - 2 obtained using neural network model are provided in Table 5.43. The R^2 values for all the curing ages are higher in neural network model than GP model as shown in Table 5.41 and Table 5.45. Figs. 5.136 - 5.141 provide comparison of the predicted results obtained using neural network, GP model and laboratory measued results. It can be observed from these figures that neural network model predicts compressive strength values very near to the actual values as compared to GP model results.

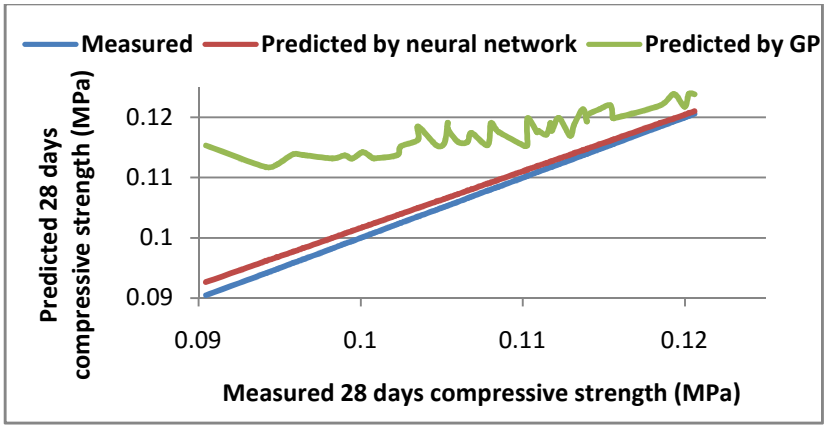

Fig. 5.136: Measured vs. predicted values of f_{c28} (using neural network and GP) for Dataset – 1

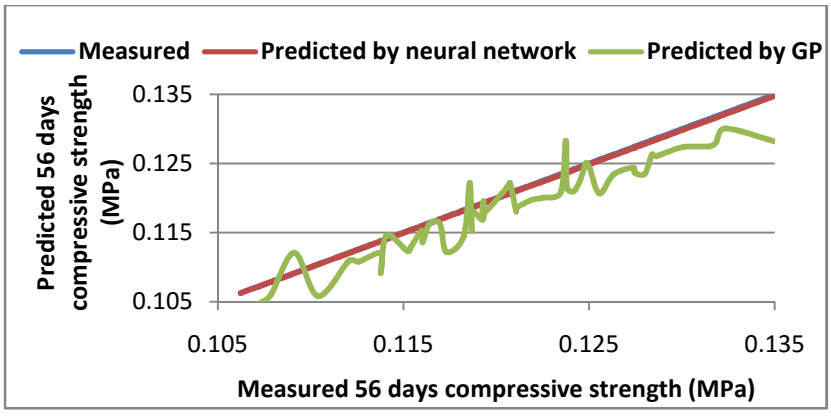

Fig. 5.137: Measured vs. predicted values of f_{c56} (using neural network and GP) for Dataset - 1

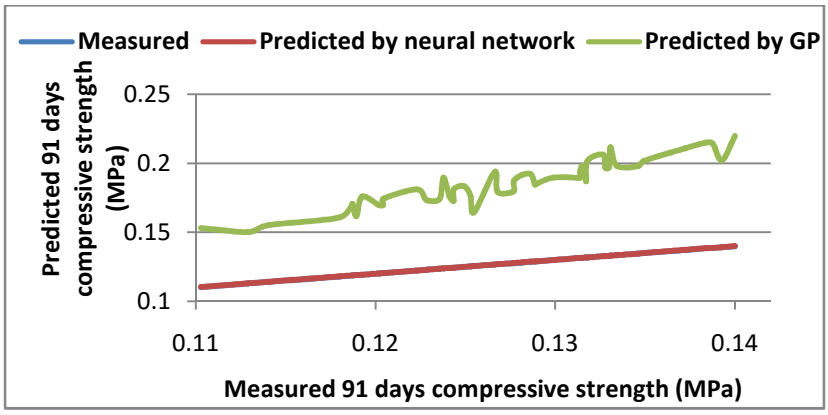

Fig. 5.138: Measured vs. predicted values of f_{c91} (using neural network and GP) for Dataset - 1

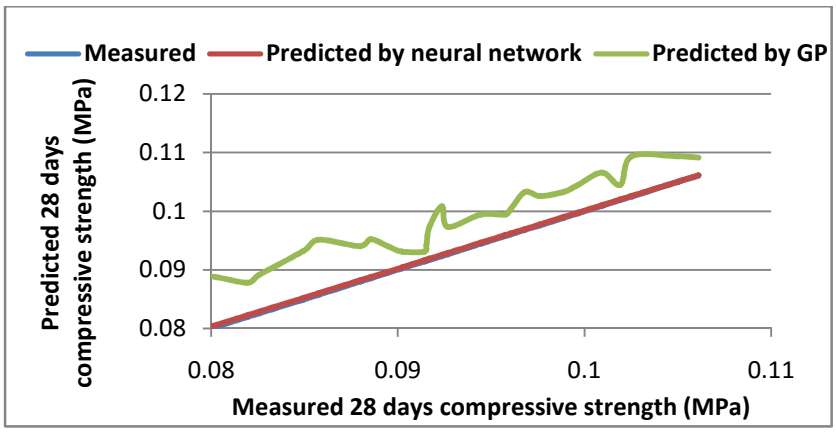

Fig. 5.139: Measured vs. predicted values of f'_{c28}(using neural network and GP) for Dataset - 2

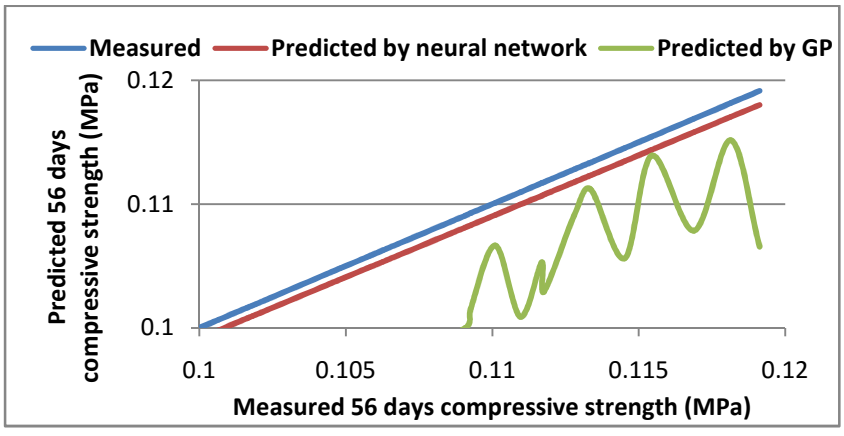

Fig. 5.140: Measured vs. predicted values of f'_{c56} (using neural network and GP) for Dataset - 2

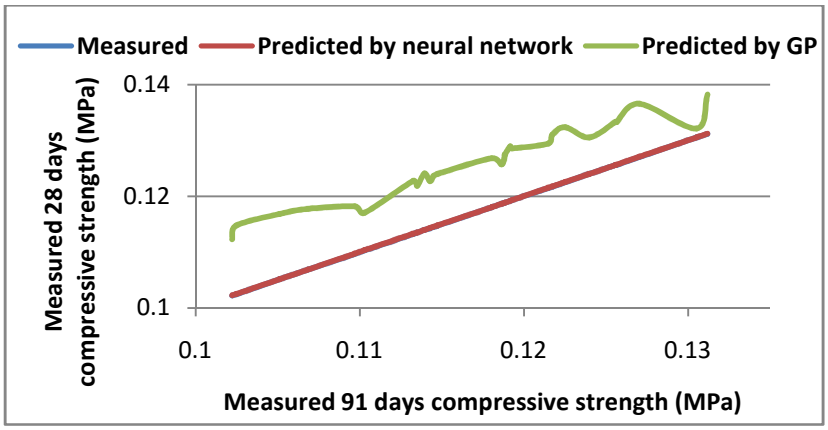

Fig. 5.141: Measured vs. predicted values of f'_{c91} (using neural network and GP) for Dataset - 2

5.6.2 Comparison with Regression model

From Section 5.6.1, it is evident that neural network with trainlm() training function has the highest predictive power for the prediction of compressive strength of concrete as compared to GP model. In the present section, proposed neural network model is compared with traditional linear regression model for the prediction of compressive strength of concrete. The values of R^2 and RMSE for Dataset - 1 and Dataset - 2 obtained using linear regression model are presented in Table 5.44. From these results, it can be observed that for Dataset - 1 and for the prediction of f_{c56} and f_{c91}, the values of R^2 are 90.8% and 87.1%, respectively, with RMSE of 0.0037 for the prediction of f_{c56} and 0.0027 for the prediction of f_{c91}, indicating a good fit. However, for the prediction of f_{c28}, the value of R^2 is 76.3% which is less than a good fit. In case of prediction of compressive strength of concrete for Dataset - 2, R^2 is greater than or equal to 91.3% for all the curing ages. The highest value of R^2 is 95.7% for the prediction of f_{c91} with RMSE as 0.0018. While comparing linear regression model results with the results of neural network model, it has been observed that proposed neural network model is predicting with R^2 greater than or equal to 99.6%, in all curing ages except for the prediction of f_{c28} which is 89.8% and is considered as a good fit and it is also much higher than linear regression model predictive value for the prediction of f_{c28} which is 76.3%.

Figs. 5.142 - 5.147 provide a comparison of the predicted results obtained using neural network model, linear regression model and actual results. It can be observed from these figures that neural network model predicts compressive strength values very near to the measured values.

Table 5.44: Results obtained from linear regression model

Linear Regression Model		Results	
Dataset	**Prediction variables**	R^2 (%)	**RMSE**
Dataset - 1	f_{c28}	76.3	0.0037
	f_{c56}	90.8	0.0022
	f_{c91}	87.1	0.0027
Dataset - 2	f'_{c28}	92.9	0.0022
	f'_{c56}	91.3	0.0028
	f'_{c91}	95.7	0.0018

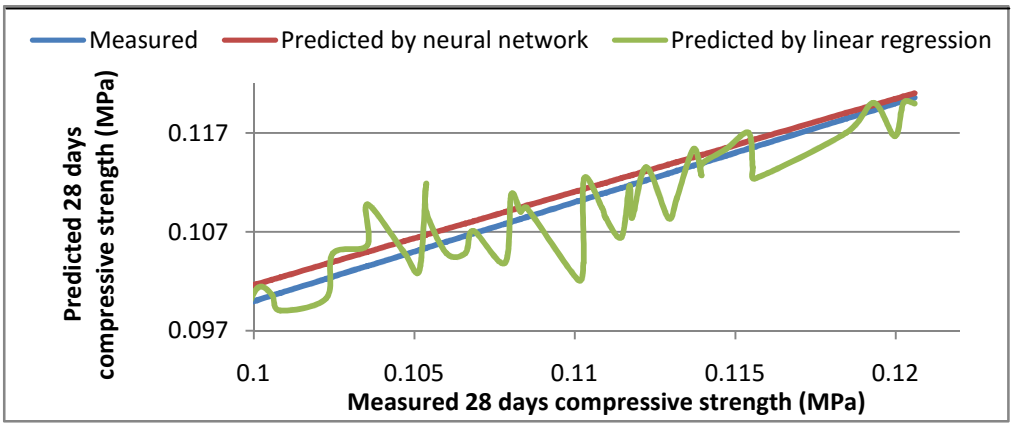

Fig. 5.142: Measured vs. predicted values of f_{c28} (using neural network and linear regression model) for Dataset - 1

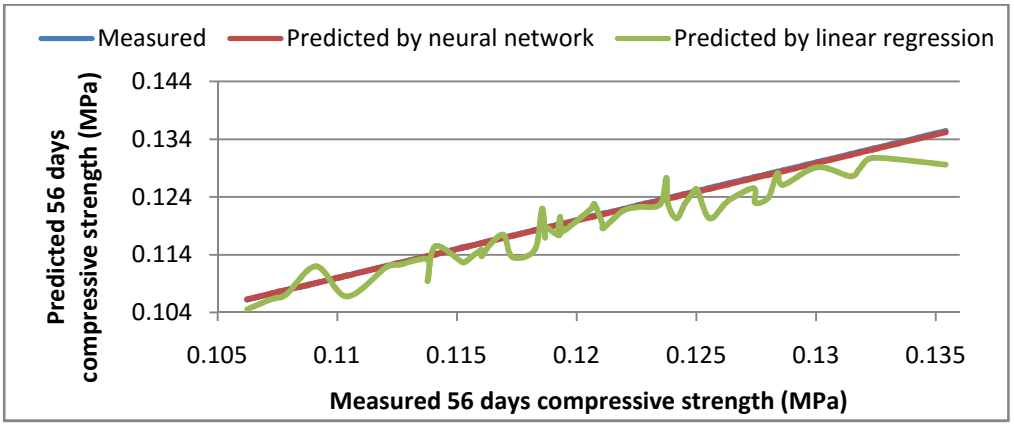

Fig. 5.143: Measured vs. predicted values of f_{c56} (using neural network and linear regression model) for Dataset - 1

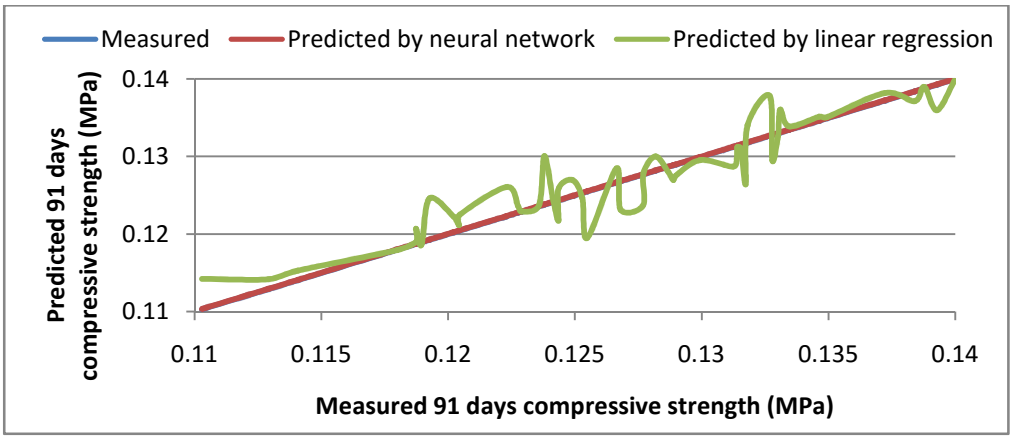

Fig. 5.144: Measured vs. predicted values of f_{c91} (using neural network and linear regression model) for Dataset - 1

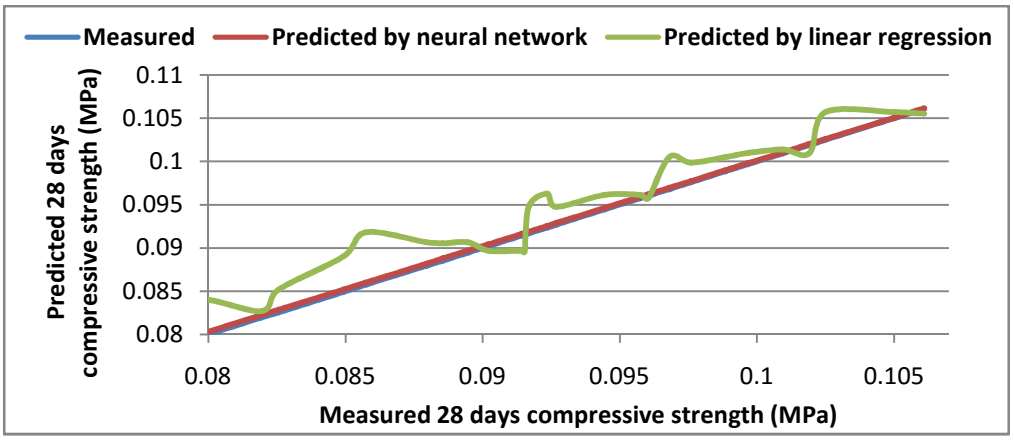

Fig. 5.145: Measured vs. predicted values of f'_{28} (using neural network and linear regression model) for Dataset - 2

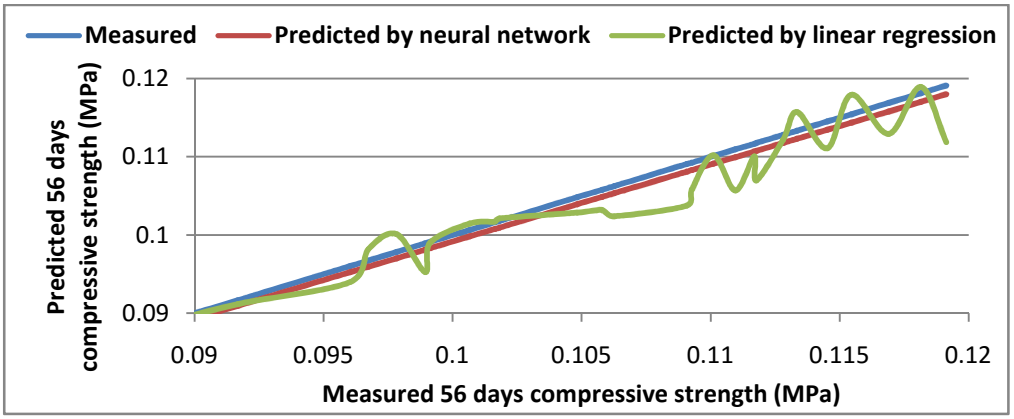

Fig. 5.146: Measured vs. predicted values of f'_{56} (using neural network and linear regression model) for Dataset – 2

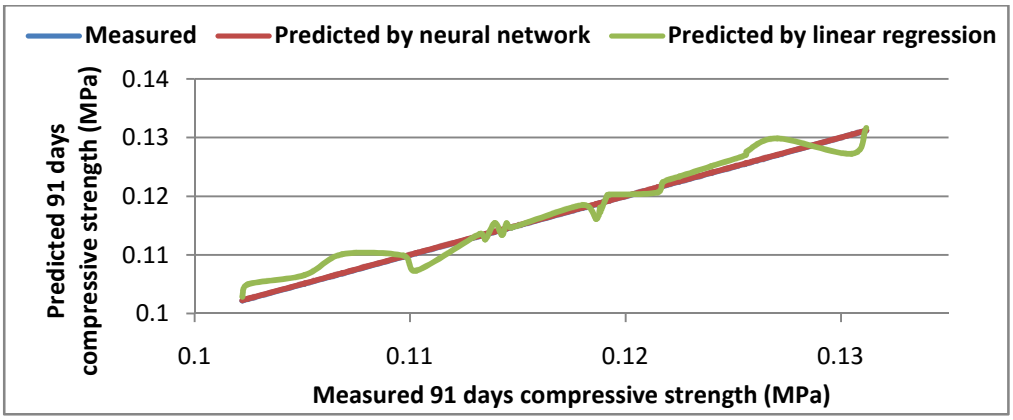

Fig. 5.147: Measured vs. predicted values of f'_{91} (using neural network and linear regression model) for Dataset - 2

5.6.3 Conclusions

Section 5.6.1 and Section 5.6.2 compared the results of neural network model with GP model and with linear regression model, respectively. These results reveal that neural network model has been predicting with higher values of R^2 as compared to GP and linear regression models. The value of R^2, when using neural network model, is greater or equal to 99.6% in all the cases except for the prediction of f_{c28} which is 89.8%, also considered as a reasonably good fit. Using proposed neural network model, for Dataset - 1, the values of R^2 are 99.8% and 100% for f_{c56} and f_{c91}, respectively and for Dataset - 2, the values of R^2 are 99.6%, 100.0%, 100.0% for the prediction of f_{c28}, f_{c56} and f_{c91}, respectively. Therefore, it has been inferred that neural network model is a good prediction model for the prediction of compressive strength of concrete.

CHAPTER 6

CONCLUSIONS AND FUTURE SCOPE

The primary objective of this study was to investigate computational aspects, including design and development, of neural network models with specific reference to prediction of compressive strength of concrete. The research findings revealed from this study are briefly summarized and certain pointers to future research in this direction have been segregated in this chapter.

6.1 Review of the results emerged from this study

In Chapter 3, we have discussed about dataset used, procedure followed for the analysis of the dataset and the variables that have highest influence on compressive strength of concrete. The data published by Kumar (2003) has been used for generating a reliable data bank on compressive strength of concrete. The data was collected in controlled laboratory conditions using different values of parameters for concrete mixes. He considered five parameters, namely, water-cementitious material ratio (*w*/*cm*), cementitious content (*c*), water content (*w*), workability and curing ages. We have carried out factor analysis to evaluate the impact of variables on compressive strength of concrete. In this work, we have applied exploratory factor analysis (EFA) with the help of Statistical Package for the Social Sciences (SPSS). The results of analysis revealed that the data adequacy of each individual variable and dataset collectively is greater than 71.20%. The *w*/*cm* ratio comes out as a leading predictor variable as it accounted for 64.70% of the total variance and every factor is sharing its maximum variability with other variables of the dataset.

The work presented in Chapter 4 comprised the development of regression models for predicting compressive strength of concrete at three stages of its curing. We have developed regression models to analyze the effect of workability on compressive strength in Section 4.3 and also to analyze the effect of FA on compressive strength in Section 4.4. We have also presented the basic linear regression model for the sake of completeness in Section 4.2 and observed that the value of R^2 is higher than 92.90% for all curing ages, and even when cement content is replaced with FA or not replaced with FA. Therefore, we have inferred that linear regression equations are well suited for all the curing ages.

In Section 4.3, we have investigated regression models for the prediction of compressive strength of concrete for varying workability to analyse the effect of workability on compressive strength. Two regression models (Model-2 and Model-3) have been developed for concrete with medium and high workability at different curing ages. The model developed by Namyong *et al.* (2004) has also been used and we have named it as Model-1 in this work. **We have inferred that nonlinear regression models (Model-2 and Model-3) provide better results for medium and high workability mixes and it is also recommended to consider f_{c28} and f_{c56} in regression equations for better prediction of f_{c91} for both medium and high workability mixes. Model-2 is best suited for f_{c28} and Model-3 gives more efficient results for f_{c91} for medium workability. Model-3 is best suited model for predicting compressive strength of high workability mixes at all ages as compared to other regression models.**

The work presented in Section 4.4 includes multiple regression models for predicting compressive strength of concrete with three different aggregate zones, *i.e.*, Zone-A, Zone-B and Zone-C with and without FA. It has been observed that this model works efficiently for Zone-A and Zone-B when cement is replaced with 15% FA and for Zone-C aggregates, when cement is not replaced by FA.

The regression models developed in Sections 4.2 - 4.4 can predict compressive strength of various mixes very efficiently. However, a variation in data affects the regression coefficients to a large extent. Therefore, we have introduced the ridge parameter in regression equations in Section 4.5. **By introducing ridge parameter in regression equations, we get more trustworthy and efficient predictive models for compressive strength of concrete that are not affected by the variations in dataset used for prediction.**

This is a well known fact that properties of concrete, including, compressive strength depend on a number of parameters. These parameters are affected by the non-homogeneous nature of concrete components, and also by the combined effects of these components. This makes the prediction task even more complex. Chapter 5 included the development of neural network and GP models for the prediction of compressive strength of concrete. We also carried out the comparisons of these models in this chapter.

In Section 5.2, we have started with preliminary experimentation with certain randomly selected network architectures on the basis of the knowledge gained from literature. We have empirically investigated 1440 neural network models for prediction of compressive strength of concrete. The aim of this experimentation was to decide for the optimum values of *lr*, *goal* and *hn*. After a critical analysis, we have finalized the values of these parameters as *lr* = 0.01, *goal* = 0.000001 and *hn* = 50. We have also excluded two training functions, namely, trainbr() and

traingda() from further experimentation as the models could not generally converge while using these functions.

In Section 5.3, we have proposed neural network model for the prediction of compressive strength of concrete. **In the development of this model, we have investigated eight training functions experimentally for network optimization and deduced that the best training algorithm for the prediction of compressive strength of concrete is 'Levenberg-Marquardt' algorithm with tan-sigmoid as the AF for hidden layer.** In Section 5.4, we have developed GP models with the objective of exploring the applicability of GP models for compressive strength prediction. We have argued that there is a well-built correlation between predicted and measured values for the data available in the dataset except for the prediction of f_{c28}. In Section 5.5, in order to test the efficacy and reliability of models, Namyong *et al*., (2004) in-situ data has also been used and it has been observed that neural network model is more reliable and provides more accurate prediction even for the in-situ dataset for concrete.

In Section 5.6, we have compared the results of neural network model with GP model and with linear regression model. **The value of R^2, when using neural network model, is greater than or also equal to 99.6% in all the cases except for the prediction of f_{c28}. We have concluded that neural network models have better prediction capabilities when compared with linear regression and GP models.**

6.2 Limitations and future scope of the study

- ✓ The future work in this direction can be initiated to further explore various emerging soft computing techniques such Support Vector Machine (SVM), Adaptive Neuro-Fuzzy Inference System (ANFIS), hybrid soft computing models on neural fuzzy and evolutionary computation technologies.
- ✓ The method of weight initialization used in the present work has concerted to the random weight initialization, further new methods of weight initialization can also be investigated.
- ✓ Different cross-validation methods need to be explored to a larger extent in order to effectively rectify the over-fitting problems inherent with the neural network models due to availability of small size of datasets for the prediction of compressive strength of concrete.
- ✓ The proposed neural network model can further be applied on a more varied dataset. For this purpose, inter-disciplinary and inter-institutional research projects need to be developed at organized research hubs.

LIST OF PUBLICATIONS

Chopra, P., Sharma, R. K. and Kumar, M., (2016). "Predicting compressive strength of concrete using artificial neural network and genetic programming". ***Advances in Materials Science and Engineering,*** Article ID 7648467, 10 pages.

Chopra, P., Sharma, R. K. and Kumar, M., (2015). "Artificial neural networks for the prediction of compressive strength of concrete". ***International Journal of Applied Sciences & Engineering,*** 13(3), 187-204.

Chopra, P., Sharma, R. K. and Kumar, M., (2014). "Predicting compressive strength of concrete for varying workability using regression models". ***International Journal of Engineering & Applied Sciences,*** 6(4), 10-22.

Chopra, P., Sharma, R. K. and Kumar, M., (2014). "Regression models for the prediction of compressive strength of concrete with & without fly ash". ***International Journal of Latest Trends in Engineering and Technology,*** 3(4), 400-406.

Chopra, P., Sharma, R. K. and Kumar, M., (2013). "Ridge regression for the prediction of compressive strength of concrete***". International Journal of Innovations in Engineering and Technology,*** 2(3), 106-111.

REFERENCES

Abdollahzadeh, A., Masoudnia, R. and Aghababaei, S., (2011). "Predict strength of rubberized concrete using artificial neural networks". ***WSEAS Transactions on Computing*****,** 10(2), 31-40.

ACI 214.3R-88, (1988). "Simplified version of the recommended practice for evaluation of strength test results, ACI committee 214 report". ***ACI Materials Journal*****,** 272-279.

Aggarwal, Y. and Aggarwal, P., (2011). "Prediction of compressive strength of SCC containing bottom ash using artificial neural networks". ***International Journal of Mathematical, Computational, Physical, Electrical and Computer Engineering*****,** 5(5), 762-767.

Akande, K. O., Owolabi, T. O., Twana, S. and Olatunji, S. O., (2014). "Performance comparison of SVM and ANN in predicting compressive strength of concrete". ***ISOR Journal of Computer Engineering***, 16(5), 88-94.

Akkurt, S., Tayfur, G. and Can, S., (2004). "Fuzzy logic model for the prediction of cement compressive strength". ***Cement and Concrete Research***, 34, 1429-1433.

Alilou, V. K. and Teshnehlab, M., (2010). "Prediction of 28-day compressive strength of concrete on the third day using artificial neural network". ***International Journal of Engineering***, 3(6), 565-576.

Alshihri, M. M., Azmy, M. A. and El Bisy, M. S., (2009). "Neural networks for predicting compressive strength of structural light weight concrete". ***Construction and Building Materials***, 23, 2214-2219.

Altendorf, C. T., Elliott, R. L., Stevens, E. W. and Stone, M. L., (1999). "Development and validation of neural network model for soil water content prediction with comparison to regression techniques". ***Transactions of the American Society of Agricultural Engineering***, 42(3), 691–699.

Anderson, J. A., (1995). "An Introduction to Neural Networks". ***The MIT Press,*** Cambridge, MA.

Atiya, A. and Ji, C., (1997). "How initial conditions affect generalization performance in large networks". ***IEEE Transactions on Neural Networks,*** 8(2), 448–451.

Basyigit, C., Akkurt, I., Kilincarslan, S. and Beycioglu, A., (2009). "Prediction of compressive strength of heavyweight concrete by ANN and FL models". ***Neural Computing and Applications***, DOI 10.1007/s00521-009-0292-9.

Baykasoglu, A., Dereli, T. and Tanis, S., (2004). "Prediction of cement strength using soft computing techniques". ***Cement and Concrete Research***, 34, 2083-2090.

Bayrak, H. and Akgül, F., (2013). "Effect of coefficients of regression model on performance prediction curves". ***International Journal of Engineering and Applied Sciences***, 5, 32-39.

Benedetti, S., Mannino, S., Sabatini, A. G. and Marcazzan, G. L., (2004). "Electronic nose and neural network use for the classification of honey". ***Apidologie, Springer Verlag***, 35 (4), 397-402.

Bishop, C. M., (1995). "Neural Networks for Pattern Recognition". ***Clarendon Press***, Oxford.

Brethour, J. R., (1994). "Estimating marbling score in live cattle from ultrasound images using pattern recognition and neural network procedures". ***Journal of Animal Sciences***, 72, 1425–1432.

Buenfeld, N. R. and Hassanein, N. M., (1998). "Prediction the life of concrete structures using neural Network". ***Proceedings of the Institution of Civil Engineers-Structures and Buildings,*** 128, 34-48.

Cheng, M. Y., Firdausi, P. M. and Proyogo, D., (2014). "High performance concrete compressive strength prediction using Genetic Weighted Pyramid Operation Tree (GWPOT)". ***Journal of Engineering Applications of Artificial Intelligence***, 29, 104-113.

Chester, D. L., (1990). "Why two hidden layers are better than one?" ***Proceedings of the International Joint Conference on Neural Networks***, Washington DC, Lawrence Erlbaum 1, 256–268.

Chong Edwin, K. P and Stanislaw, H. Z. (2004). "An introduction to optimization". ***Wiley,*** Singapore.

Chore, H. S., and Shaelke, N. L., (2013). "Prediction of compressive strength of concrete using multiple regression model". ***An International Journal of Structural Engineering and Mechanics***, 45(6), 837-851.

Chou, J., Tsai, C., Pham, A. and Lu, Y., (2014). "Machine learning in concrete strength simulations: Multi-nation data analytics". ***Construction and Building Materials***, 73, 771-780.

Chou, J. S., Farfora, M., Chiu, C. K. and Altahawa, I., (2011). "Optimizing prediction accuracy of concrete compressive strength based on a comparison of data mining techniques". ***Journal of Computing in Civil Engineering***, 25(3), 242-253.

Coakley, K. J., (1991). "A cross-validation procedure for stopping the EM algorithm and deconvolution of neutron depth profiling spectra". ***IEEE Transactions on Nuclear Science,*** 38(1), 9–15.

Dadkhah, R. and Esfahani, N. M., (2013). "Application of genetic programming to modelling of uniaxial compressive strength". ***Middle-East Journal of Scientific Research***, 15(6), 840-845.

Dantas, A. T. A., Leite, M. B. and Nagahama, K. D. J., (2013). "Prediction of compressive strength of concrete containing construction and demolition waste using artificial neural networks". ***Construction and Building Materials***, 38, 717-722.

Dara, R. A. and Kamel M., (2004). "Sharing training patterns in neural network ensembles". ***Proceedings of the IEEE International Conference on Neural Networks,*** 2, 157–1161.

Dayhoff, J. E., (Editor), (1990). "Neural network architectures: an introduction". ***Van Nostrand Reinhold,*** NY.

Demuth, H. B. and Beale, M. (2004). "Neural network toolbox for use with matlab, user's guide". ***The MathWorks Inc.,*** Natick, MA.

Desai, V. S. and Bharti, R., (1998). "A comparison of linear regression and neural network methods for predicting excess returns on large stocks". ***Annals of Operations Research***, 78, 127–163.

Diab, A. M., Abdelwahab, A. M., Elyamany, H. E. and Abdelmoty, A. M., (2012). "Guidelines in compressive strength assessment of concrete modified with silica fume due to magnesium sulfate attack". ***Construction and Building Materials***, 36, 311–318.

Diab, M. A., Elyamany, E. H., Elmoaty, A. E. M .A. and Shalan, H. A., (2014). "Prediction of concrete compressive strength due to long term sulfate attack using neural network". ***Alexandria Engineering Journal,*** 53(3), 627-642.

Doan, C. D. and Liong, S. Y., (2004). "Generalization for multilayer neural network: Bayesian regularization or early stopping". ***Proceedings of 2nd Conference of the Asia Pacific Association of Hydrology and Water Resources,*** Singapore, July 5-8.

Dowell, F. E., (1993). "Neural network classification of undamaged and damaged peanut kernels using spectral data". ***American Society of Agricultural Engineering,*** 93-3050.

Drummond, S. T., Sudduth, K. A., Joshi, A., Birrell, S. J. and Kitchen, N. R., (2003). "Statistical and neural methods for site-specific yield prediction". ***Transactions of the American Society of Agricultural Engineering***, 46(1), 5–14.

Duch, W. and Jankowski, N., (1999). "Survey of neural transfer functions". ***Neural Computing Surveys 2,*** 163–212.

Erdal, M., (2009). "Prediction of the compressive strength of vacuum processed concretes using artificial neural network and regression techniques". ***Scientific Research and Essay 4,*** 1057-1065.

Erenturk, K., Erenturk, S. and Tabil, L. G., (2004). "A comparative study for the estimation of dynamical drying behavior of Echinacea angustifolia: regression analysis and neural network". ***Computers and Electronics in Agriculture***, 45, 71–90.

Fang, Q., Biby, G., Haque, E., Hanna, M. A. and Spillman, C. K., (1998). "Neural network modeling of physical properties of ground wheat". ***Cereal Chemistry***, 75(2), 251–253.

Fazel Zarandi, M. H., Turksen, I. B., Sobhani, J. and Ramezanianpour, A. A., (2008). "Fuzzy polynomial neural networks for approximation of the compressive strength of concrete". ***Applied Soft Computing***, 8, 488-498.

Ferentinos, K. P., (2005). "Biological engineering applications of feedforward neural networks designed and parameterized by genetic algorithms". ***Neural Networks***, 18, 934–950.

Ferentinos, K. P. and Albright, L. D., (2002). "Predictive neural network modeling of pH and electrical conductivity in deep-trough hydroponics". ***Transactions of the American Society of Agricultural Engineering***, 45(6), 2007–2015.

Flood, A. and Kartam, N. (1994). "Neural networks in civil engineering I: principles and understanding". ***Journal of Computing in Civil Engineering,*** 8(2), 149-162.

Fogel, D. B., (1991). "An information criterion for optimal neural network selection". ***IEEE Transactions on Neural Networks,*** 2(5), 490–497.

Freeman, J. A., (1993). "Back-propagation in a neural network". ***AI Expert-Neural Network special report,*** 55–63.

Freeman, J. A. and Skapura, D. M., (2004). "Neural Networks-Algorithms, Applications and Programming Techniques". 9th Indian Reprint. ***Pearson Education (Singapore) Pte. Ltd.,*** Indian Branch, Delhi.

Fujita, O., (1998). "Statistical estimation of the number of hidden units for feedforward neural networks". ***Neural Networks***, 11(5), 851–859.

Ghaboussi J., Garrett J. H. and Wu X., (1991). "Knowledge- based modelling of material behaviour with neural networks". ***Journal of Engineering Mechanics***, 117, 132-153.

Ghedira, H. and Bernier, M., (2004). "The effect of some internal neural network parameters on SAR texture classification performance". ***Proceedings of the IEEE International Geosciences and Remote Sensing Symposium,*** Anchorage, Alaska 6, 3845–3848.

Ghodratnamaa, R., Moghaddam,T. and Babolic, A., (2013). "Comparing three proposed meta-heuristics to solve a new p-hub location Problem", ***International Journal of Engineering***, 26, 1043-1058.

Glass, G. K., Hassanein, N. M. and Buenfeld N. R., (1997). "Neural network modelling of choride binding". ***Magazine of Concrete Research,*** 49, 323-335.

Gnana Sheela, K. and Deepa, S. N., (2013). "Review on methods to fix number of hidden neurons in neural networks". ***Mathematical Problems in Engineering***. Article ID 425740. doi:10.1155/2013/425740.

Goh, A. T. C., (1995). "Prediction of ultimate shear strength of deep beams using neural networks". ***ACI Structural Journal***, 92, 28-32.

Gorphade, V. G., Rao, H. S. and Beulah, M., (2014). "Development of genetic algorithm based neural network model for predicting workability and strength of high performance concrete". ***International Journal of Inventive Engineering and Sciences,*** 2(6), 1-8.

Grzesiak, W., Błaszczyk, P. and Lacroix, R., (2006). "Methods of predicting milk yield in dairy cows - predictive capabilities of wood's lactation curve and artificial neural networks (ANNs)". ***Computers and Electronics in Agriculture***, 54, 69–83.

Grzesiak, W., Lacroix, R., Wojcik, J. and Blaszczyk, P., (2003). "A comparison of neural network and multiple regression predictions for 305-day lactation yield using partial lactation records". ***Canadian Journal of Animal Science***, 83, 307–310.

Guyon, I. and Elisseeff, A., (2003). "An introduction to variable and feature selection". ***Journal of Machine Learning Research***, 3, 1157–1182.

Hagan, M., Demuth, T. and Beale, M., (2004). "Neural network design". ***PWS Publishing Co.,*** Boston, MA, USA.

Haj-Ali, R. M., Kurtis, K. E. and Sthapit, A. R., (2001). "Neural network modeling of concrete expansion during long-term sulfate exposure". ***Journal of ACI Materials***, 98, 36–43.

Hakim, S. J., Noorzaei, J., Jaafar, M. S., Jameel, M. and Mohammadhassani, M., (2011). "Application of artificial neural networks to predict compressive strength of high strength concrete". ***International Journal of Physical Sciences***, 6(5), 971-981.

Hasan, M. M. and Kabir, A., (2011). "Prediction of compressive strength of concrete from early age test result". ***Proceedings of 4th Annual Paper Meet and 1st Civil Engineering Congress, Dhaka, Bangladesh,*** 1-7.

Hassoun, M. H., (1995). "Fundamentals of artificial neural networks". ***The MIT Press,*** Cambridge, MA.

Heald, C. W., Kim, T., Sischo, W. M., Cooper, J. B. and Wolfgang, D. R., (2000). "A computerized mastitis decision aid using farm-based records: an artificial neural network approach". ***Journal of Dairy Science***, 83, 711–720.

Heshmati, A. A. R., Salehzade, H., Alavi, A. H., Gandomi, A. H. and Abadi, M. M., (2010). "A multi expression programming application to high performance concrete". ***World Applied Sciences Journal***, 11(11), 1458- 1466.

Hessami, M., Anctil, F. and Viau, A. A., (2004). "Selection of an artificial neural network model for the post-calibration of weather radar rainfall estimation". ***Journal of Data Science***, 2, 107–124.

Hong-Guang, Ni. and Ji-Zong, W., (2000). "Prediction of compressive strength of concrete by neural networks". ***Cement and Concrete Research***, 30, 1245-1250.

Hornik, K., Stinchcombe, M. and White, H., (1989). "Multilayer feed-forward networks are universal approximators". ***Neural Networks***, 2, 359–366.

Hunter, D., Yu, H., Pukish, M. S., Kolbusz, J. and Wilamowski, B. M., (2012). "Selection of proper neural network sizes and architectures: a comparative study". ***IEEE Transactions on Industrial Informatics***, 8(2), 228–240.

Hwang, K., Noguchi, T. and Tomosawa, F., (2004). "Prediction model of compressive strength development of fly-ash concrete". ***Cement Concrete Research,*** 34, 2269-2276.

IS: 383-1970 (Second Revision), "Specifications for coarse and fine aggregates from natural resources for concrete".

IS: 516 – 1959, "Methods for tests for strength of concrete".

IS: 8112-1989, "Specifications for high strength ordinary portland cement".

Jamil, M., Zain, M. F. M. and Basri, H. B., (2009). "Neural network simulator model for optimization in high performance concrete mix design". **European Journal of Scientific Research**, 34, 61-68.

Jinchuan, K. and Xinzhe, L., (2008). "Empirical analysis of optimal hidden neurons in neural network modeling for stock prediction". ***Proceedings of the Pacific-Asia Workshop on Computational Intelligence and Industrial Application***, 2, 828–832.

Kamalloo, A., Ganjkhanlou, Y., Aboutalebi, S. H. and Nouranian, H., (2010). "Modeling of compressive ctrength of etakaolin based geopolymers by the use of artificial neural network". ***International Journal of Engineering***, 23,145-152.

Karahan, O., Tanyildizi, H. and Atis, C.D., (2008). "An artificial neural network approach for prediction of long bterm strength properties of steel fiber reinforced concrete containing fly ash". ***Journal of Zhejiang University Science A***, 9(11), 1514-1523.

Karsoliya, S., (2012). "Approximating Number of Hidden layer neurons in multiple hidden layer BPNN architecture". ***International Journal of Engineering Trends and Technology***, 3(6), 714-717.

Kecman, V., (2001). "Learning and soft computing: support vector machines, neural networks and fuzzy logic models. ***The MIT Press,*** Cambridge, MA.

Kewalramani, M. A. and Gupta, R., (2006). "Concrete compressive strength prediction using ultrasonic pulse velocity through artificial neural networks". ***Automation in Construction 15***, 374-379.

Khan, S. U., Ayub, T. and Rafeeqi, S. F. A., (2013). "Prediction of compressive strength of plain concrete confined with ferrocement using Artificial Neural Network (ANN) and comparison

with existing mathematical models". ***American Journal of Civil Engineering and Architecture***, 1(1), 7-14.

Kheder, G. F., Al-Gabban, A. M. and Suhad, M. A., (2003). "Mathematical model for the prediction of cement compressive strength at the ages of 7 and 28 days within 24 hours". ***Materials and Structure***, 36, 693-701.

Knight, K., (1990). "Connectionist ideas and algorithms". ***Communications of the ACM***, 33(11), 59–74.

Kominakis, A. P., Abas, Z., Maltaris, I. and Rogdakis, E., (2002). "A preliminary study of the application of artificial neural networks to prediction of milk yield in dairy sheep". ***Computers and Electronics in Agriculture***, 35, 35–48.

Koza J. R., (1992). "Genetic programming: on the programming of computers by means of natural selection". ***The MIT Press,*** Cambridge, MA.

Kumar, M., (2003). "Reliability based design of structural elements". ***Thesis submitted for degree of doctor of philosophy,*** T.I.E.T, Patiala

Kwok, T.Y. and Yeung, D.Y., (1995). "Efficient cross-validation for feedforward neural networks". ***Proceedings of the IEEE International Conference on Neural Networks***, 5, 2789–2794.

Lacroix, R., Salehi, F., Yang, X. Z. and Wade, K. M., (1997). "Effects of data preprocessing on the performance of artificial neural network for dairy yield prediction and cow culling classification". ***Transactions of the American Society of Agricultural Engineering***, 40(3), 839–846.

Lee, S.C., (2003). "Prediction of concrete strength using artificial neural networks". ***Engineering Structures*** , 25, 849-857.

Lee, Y., Oh, S., Kim and M. W., (1991). "The effect of initial weights on premature saturation in back-propagation learning". ***Proceedings of the IEEE International Joint Conference on Neural Networks***, 1, 765–770.

Li, J. Y., Chow, T. W. S. and Yu, Y. L., (1995). "Estimation theory and optimization algorithm for the number of hidden units in the higher-order feedforward neural network". ***Proceedings of the IEEE International Conference on Neural Networks***, 3, 1229–1233.

Lippmann, R. P., (1987). "An introduction to computing with neural nets". ***IEEE Acoustics, Speech and Signal Processing Magazine***, 4(2), 4–22.

Liu, J., Goering, C. E. and Tian, L., (2001). "A neural network for setting target corn yields". ***Transactions of the American Society of Agricultural Engineering***, 44(3), 705–713.

Madandoust, R., Bungey, J.H. and Ghavidel, R., (2012). "Prediction of the concrete compressive strength by means of core testing using GMDH-type neural network and ANFIS models". ***Computational Materials Science***, 51(1), 261-272.

Medler, D. A., (1998). "A brief history of connectionism". ***Neural Computing Surveys***, 1, 61–101.

Mehta, P. K. and Monteiro, P. J. M., (2006). "Concrete, microstructure, properties and materials". 3rd Edition. ***McGraw-Hill***, USA.

Michie, D., Spiegelhalter, D. J. and Taylor, C. C., (Editors) (1994). "Machine learning, neural and statistical Classification". Available at: ***http://www.maths.leeds.ac.uk/~charles/statlog/whole.pdf***.

Moody, J. E. and Yarvin, N., (1992). "Networks with learned unit response functions". ***Advances in Neural Information Processing Systems***, 4, 1048–1055.

Mukherjee, A. and Deshpande, J. M., (1995). "Modeling initial design process using artificial neural networks". ***ASCE Journal of Computing in Civil Engineering***, 9(3), 194–200.

Muthupriya, P., Subramanian, K. and Vishnuram, B. G., (2011). "Prediction of compressive strength and durability of high performance concrete by artificial neural networks". ***International Journal of Optimization in Civil Engineering***, 1, 189-209.

Namyong, J., Sangchun, Y., and Hongburm, C., (2004). "Prediction of compressive strength of In-situ concrete based on mixture proportions". ***Journal of Asian Architecture and Building Engineering,*** 3, 9-15.

Nikoo, M., Moghadam, F. T. and Sadowski, L., (2015). "Prediction of concrete compressive strength by evolutionary artificial neural networks". ***Advances in Materials Science and Engineering***, Article ID 849126, 8 pages. doi:10.1155/2015/849126.

Noorzaei, J., Hakim, S. J. S., Jaafar, M. S. and Thanoon, W. A. M., (2007). "Development of artificial neural networks for predicting concrete compressive strength". ***International Journal of Engineering and Technology***, 4, 141-153.

Ozcan, F., Cengiz, D. A., Karahan, O., Uncuoglu, E. and Tanyildizi, H., (2009). "Comparison of artificial neural network and fuzzy logic models for prediction of long term compressive strength of silica fume concrete". ***Advanced in Engineering Software***, 20, 856-863.

Oztas, A., Pala, M., Ozbay, E., Kanca, E., Caglar, N. and Bhatti, M.A., (2006). "Prediciting the compressive strength and slump of high strength concrete using neural network". ***Construction and Building Materials***, 20, 769-775.

Ozturan, M., Kutlu, B. and Ozturan, T., (2008). "Comparison of concrete strength prediction techniques with artificial neural network approach". ***Building Research Journal***, 56, 23-36.

Pala, M., Ozbay, E., Oztas, A. and Yuce, M. I., (2007). "Appraisal of long term effects of fly ash and silica fume on compressive strength of concrete by neural networks". ***Construction and Building Materials***, 21, 384-394.

Pietersma, D., Lacroix, R., Lefebvre, D. and Wade, K. M., (2003). "Performance analysis for machine-learning experiments using small data sets". ***Computers and Electronics in Agriculture***, 38, 1–17.

Popovics, S. and Popovics, J.S., (1996). "Novel aspects in computerization of concrete proportioning". ***Concrete International***, 54-58.

Popovics, S. and Ujhelyi, J., (2008). "Contribution to the concrete strength versus water-cement ratio relationship". ***Journal of Materials and Civil Engineering***, 20,459-463.

Popovics, S., (1990). "Analysis of concrete strength versus water-cement ratio relationship". ***ACI Material Journal***, 87, 517-529.

Prasad, B. K., Eskandari, H. and Reddy, V., (2009). "Prediction of compressive strength of SCC and HPC with high volume fly ash using ANN". ***Construction and Building Materials***, 23, 117-128.

Rajamane, N. P., Peter, J. A. and Ambily, P. S., (2007). "Prediction of compressive strength of concrete with fly ash as sand replacement material". ***Cement and Concrete Composites***, 29, 218-223.

Rasa, E., Ketabchi, H. and Afshar, M. H., (2009). "Predicting density and compressive strength of concrete cement paste containing silica fume using artificial neural networks". ***Sharif University of Technology***, 16, 33-42.

Refenes, A. N., (1995). "Neural network design considerations". ***John Wiley***, NY.

Rumelhart, D. E., Widrow, B. and Lehr, M. A., (1994). "The basic ideas in neural networks". ***Communications of the ACM,*** 37(3), 87–92.

Salehi, F., Lacroix, R. and Wade, K. M., (1998a). "Improving dairy yield predictions through combined record classifiers and specialized artificial neural networks". ***Computers and Electronics in Agriculture***, 20, 199–213.

Sanad, A., and Saka, M. P., (2001). "Prediction of ultimate shear strength of reinforced-concrete deep beams using neural networks". ***Journal of Structural Engineering,*** ASCE 127(7), 818–828.

Saridemir, M., (2009). "Prediction of compressive strength of concretes containing metakaolin and silica fume by artificial neural networks". ***Advances in Engineering Software***, 40, 350-355.

Saridemir, M., (2010). "Genetic programming approach for prediction of compressive strength of concretes containing rice husk ash". ***Construction and Building Materials***, 24, 1911-1919.

Sarle, W. S., (Editor), (1997). "How many hidden layers should I use?" Available at: ***http://www.faqs.org/faqs/ai-faq/neural-nets/part3/section-9.html***.

Sayed Ahmed, M. S., (2012). "Statistical modelling and prediction of compressive strength of concrete". ***Concrete Research Letters***, 3(2), 452- 458.

Sharma, A. K., Sharma, R. K. and Kasana, H. S., (2007). "Prediction of first lactation 305-day milk yield in karan fries dairy cattle using ANN modeling". ***Applied Soft Computing***, 7, 1112-1120.

Shibata, K. and Ikeda, Y., (2009). "Effect of number of hidden neurons on learning in large-scale layered neural networks". ***Proceedings of the ICROS-SICE International Joint Conference***, 5008–5013.

Sidda Reddy, B., Kumar, S. and Reddy, V. K., (2013). "Prediction of deflection and stresses of laminated composite plate with an artificial neural network aid". ***International Journal of Applied Science and Engineering***, 11(4), 393-413

Simpson, P. K., (Editor), (1996). "Neural networks, theory, technology, and applications". ***IEEE Inc.,*** NY.

Skoundrianos, E. N. and Tzafestas, S. G., (2004). "Modeling and FDI of dynamic discrete time systems using a MLP with a new sigmoidal activation function". ***Journal of Intelligent Systems and Robotics***, 41, 19–36.

Sontag, E. D., (1992). "Feedback stabilization using two-hidden-layer nets". ***IEEE Transactions on Neural Networks,*** 3, 981–990.

Steven, C., Raymond, C. and Canale, P., (2002). "Numerical methods for engineers with personal computer applications". ***McGraw Hill,*** New York.

Sumpter, B. G. and Donald, W. N., (1996). "On the design, analysis, and characterization of materials using computational neural networks". ***Annual Reviews of Materials Science 26,*** 223–277.

Svozil, D., Kvasnicka, V. and Prospichal J., (1997). "Introduction to multilayer feedforward neural networks-tutorial". ***Chemometrics and Intelligent Laboratory Systems***, 39, 43–62.

Tamura, S. and Tateishi, M., (1997). "Capabilities of a four-layered feedforward neural network: four layers versus three". ***IEEE Transactions on Neural Networks***, 8(2), 251–255.

Thamma, P. and Barai, S.V., (2009). "Prediction of compressive strength of concrete using genre expression programming". ***Applications of Soft Computing***, 58, 203-212.

Tiryaki, S. and Aydin, A., (2014). "An artificial neural network model for predicting compressive strength of heat treated woods and comparison with a multiple linear regression model". ***Construction and Building Materials***, 62, 102-108.

Topcu, I. B. and Saridemir, M., (2008). "Prediction of compressive strength of concrete containing fly ash using artificial neural networks and fuzzy logic". ***Computational Materials Science***, 41, 305-311.

Tsivilis, S. and Parissakis, G., (1995). "A mathematical-model for the prediction of cement strength". ***Cement Concrete Research***, 25, 9-14.

Wankhade, M. W. and Kambekar, A. R., (2013). "Prediction of compressive strength of concrete using artificial neural network". ***International Journal of Scientific Research and reviews***, 2(2), 11-26.

Wasserman, P. D., (1993). "Advanced methods in neural computing". ***Van Nostrand,*** Reinhold, NY.

Wessels, L. F. A. and Barnard, E., (1992). "Avoiding false local minima by proper initialization of connections". ***IEEE Transactions of Neural Networks***, 3(6), 899–905.

Wu, J., (1994). "Neural network and simulation methods". ***Marcel Dekker,*** Inc., NY.

Wu, S. S., Li, B. Z., Yong, J. G. and Shukla, S. K., (2010). "Predictive modelling of high-performance concrete with regression analysis". ***IEEE International Conference on Industrial Engineering and Engineering Management***, 1009-1013.

Xu, S. and Chen, L., (2008). "A novel approach for determining the optimal number of hidden layer neurons for FNN's and its application in data mining". ***Proceedings of the 5th International Conference on Information Technology and Applications,*** 683–686.

Yang, X. Z., Lacroix, R. and Wade, K. M., (1999). "Neural detection of mastitis from dairy herd improvement records". ***Transactions of the American Society of Agricultural Engineering,*** 42(4), 1063–1071.

Yeh, I. C., (1998). "Modeling of strength of high-performance concrete using artificial neural networks". ***Cement and Concrete Research,*** 28(12), 1797-1808.

Yu, X., Loh, N. K., Jullien, G. A. and Miller, W. C., (1993). "Comparisons of four learning algorithms for training the multilayer feedforward neural networks with hard-limiting neurons". ***IEEE Proceedings of the Canadian Conference on Electrical and Computer Engineering***, 1, 477–480.

Yuan, H. C., Xiong, F. L. and Huai, X. Y., (2003). "A method for estimating the number of hidden neurons in feed-forward neural networks based on information entropy". ***Computers and Electronics in Agriculture***, 40, 57–64.

Zain, M. F. M and Abd, S. M., (2009). "Multiple regressions model for compressive strength prediction of high performance concrete". ***Journal of Applied Sciences***, 9, 155-160.

Zain, M. F. M., Mahmud, H. B., Ilham, A. and Faizal, M., (2002). "Prediction of splitting tensile strength of high-performance concrete". ***Cement Concrete Research***, 32, 1251-1258.

Zelic, J., Rusic, D. and Krstulovic, R., (2004). "A mathematical model for prediction of compressive strength in cement-silica fume blends". ***Cement Concrete Research***, 34, 2319-2328.

Zhang, Z., Ma, X. and Yang, Y., (2003). "Bounds on the number of hidden neurons in three-layer binary neural networks". ***Neural Networks***, 16 (7), 995–1002.